**Dr. SNS RAJALAKSHIMI COLLEGE OF ARTS AND SCIENCE**



 VIRTUALIZATION TECHNOLOGY FUNDAMENTALS

 UNIT-II

UNIT-II

Understanding Virtual machines

# 2.1 Understanding Virtual Machines:

A virtual machine is a software computer that, like a physical machine, runs an operating system and applications. A virtual machine uses the physical resources of the physical machine on which it runs, which is called the host system. Virtual machines have virtual devices that provide the same functionality as physical hardware, but with the additional benefits of portability, manageability, and security.

A virtual machine has an operating system and virtual resources that you manage in much the same way that you manage a physical computer. For example, you install an operating system in a virtual machine in the same way that you install an operating system on a physical computer. You must have a CD-ROM, DVD, or ISO image that contains the installation files from an operating system vendor.

Multiple guests can run on the same [Oracle VM Server](https://docs.oracle.com/cd/E50245_01/E50249/html/go01.html#gloss-server). A virtual machine is a guest operating system and its associated application software. For the sake of simplicity, we use the term virtual machine to encompass domain, guest and virtual machine. They are synonymous with each other and may be used interchangeably.

An operating system installed in a virtual machine is known as a guest operating system. Oracle VM supports a variety of guest operating systems including Linux, Oracle Solaris and Microsoft Windows™.

**2.2. Describing virtual machine:**

Multiple guests can run on the same [Oracle VM Server](https://docs.oracle.com/cd/E50245_01/E50249/html/go01.html#gloss-server). A virtual machine is a guest operating system and its associated application software. For the sake of simplicity, we use the term virtual machine to encompass domain, guest and virtual machine. They are synonymous with each other and may be used interchangeably.

An operating system installed in a virtual machine is known as a guest operating system. Oracle VM supports a variety of guest operating systems including Linux, Oracle Solaris and Microsoft Windows™.

In computing, a "**virtual machine**" (**VM**) is the [virtualization](https://en.wikipedia.org/wiki/Virtualization) or [emulation](https://en.wikipedia.org/wiki/Emulator) of a [computer system](https://en.wikipedia.org/wiki/Computer_system). Virtual machines are based on [computer architectures](https://en.wikipedia.org/wiki/Computer_architecture) and provide the functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination of the two. Virtual machines differ and are organized by their function, shown here:

* [*System virtual machines*](https://en.wikipedia.org/wiki/System_virtual_machine) (also called [full virtualization](https://en.wikipedia.org/wiki/Full_virtualization) VMs) provide a substitute for a real machine. They provide the functionality needed to execute entire [operating systems](https://en.wikipedia.org/wiki/Operating_system). A [hypervisor](https://en.wikipedia.org/wiki/Hypervisor) uses [native execution](https://en.wikipedia.org/wiki/Native_code) to share and manage hardware, allowing for multiple environments that are isolated from one another yet exist on the same physical machine. Modern hypervisors use [hardware-assisted virtualization](https://en.wikipedia.org/wiki/Hardware-assisted_virtualization), with virtualization-specific hardware features on the host CPUs providing assistance to hypervisors.
* **Process virtual machines**  are designed to execute computer programs in a platform-independent environment.

Some virtual machine emulators, such as [QEMU](https://en.wikipedia.org/wiki/QEMU) and [video game console emulators](https://en.wikipedia.org/wiki/Video_game_console_emulator), are designed to also emulate (or "virtually imitate") different system architectures, thus allowing execution of software applications and operating systems written for another [CPU](https://en.wikipedia.org/wiki/CPU) or architecture. [Operating-system-level virtualization](https://en.wikipedia.org/wiki/Operating-system-level_virtualization) allows the resources of a computer to be partitioned via the [kernel](https://en.wikipedia.org/wiki/Kernel_%28operating_system%29). The terms are not universally interchangeable.

**2.3. Examine CPU in Virtual machine:**

The number of CPU cores in a virtual machine is an important factor in determining the performance of the machine. It is important to know how many cores are available in a virtual machine before you decide to purchase it. In this article, we will discuss how to check the number of CPU cores in a virtual machine.

**2.3.1. Checking CPU Cores in the Virtual Machine Settings**

The first step to check the number of CPU cores in a virtual machine is to check the settings of the virtual machine. Most virtual [machines](https://www.alibabacloud.com/product/machine-learning/pricing) have a [settings page](https://www.alibabacloud.com/help/en/doc-detail/118174.htm) where you can view the number of CPU cores available. This page will also show other [information](https://www.alibabacloud.com/solutions/financial/supply_chain_financing) such as the amount of [RAM](https://www.alibabacloud.com/product/ram) and storage available. To access the settings page, you will need to log into the virtual machine and then navigate to the settings page. Once you have accessed the settings page, you will be able to view the number of CPU cores available. This number will be displayed in the “CPU Cores” section of the settings page. You can also view the amount of RAM and storage available in the same section.

**2.3.2. Checking CPU Cores in the System Information**

Another way to check the number of CPU cores in a virtual machine is to check the [system information](https://www.alibabacloud.com/help/en/doc-detail/406284.htm). This information can be accessed by opening the “System Information” window in the virtual machine. This window will show the number of CPU cores available in the virtual machine. The [system information window](https://www.alibabacloud.com/solutions/financial/supply_chain_financing) will also show other information such as the amount of RAM and storage available. You can also view the type of processor and the speed of the processor in the system information window.

**2.3.3 Checking CPU Cores in the Task Manager**

The [task manager](https://www.alibabacloud.com/help/en/doc-detail/422866.htm) is another way to check the number of CPU cores in a virtual machine. The task manager can be accessed by opening the “Task Manager” window in the virtual machine. This window will show the number of CPU cores available in the virtual machine. The task manager will also show other information such as the amount of RAM and storage available. You can also view the type of processor and the speed of the processor in the task manager window.

**2.3.4. Checking CPU Cores in the**[System Properties](https://www.alibabacloud.com/product/chatapp)

The [system properties window](https://www.alibabacloud.com/product/chatapp) is another way to check the number of CPU cores in a virtual machine. This window can be accessed by opening the “System Properties” window in the virtual machine. This window will show the number of CPU cores available in the virtual machine. The system properties window will also show other information such as the amount of RAM and storage available. You can also view the type of processor and the speed of the processor in the system properties window.

**2.3.5. Checking CPU Cores in the**[BIOS](https://www.alibabacloud.com/blog/597367)

The BIOS is another way to check the number of CPU cores in a virtual machine. The BIOS can be accessed by entering the BIOS setup menu in the virtual machine. This menu will show the number of CPU cores available in the virtual machine. The BIOS will also show other information such as the amount of RAM and storage available. You can also view the type of processor and the speed of the processor in the BIOS setup menu.

### 2.3.6. Checking CPU Cores in the Operating System

The [operating system](https://www.alibabacloud.com/help/en/doc-detail/434128.htm) is another way to check the number of CPU cores in a virtual machine. The operating system can be accessed by opening the “System Information” window in the virtual machine. This window will show the number of CPU cores available in the virtual machine. The operating system will also show other information such as the amount of RAM and storage available. You can also view the type of processor and the speed of the processor in the [operating system window](https://www.alibabacloud.com/product/chatapp).

**2.3.7. Checking CPU Cores in the Virtual Machine Software**

The virtual [machine software](https://www.alibabacloud.com/product/machine-learning/pricing) is another way to check the number of CPU cores in a virtual machine. The virtual machine software can be accessed by opening the “Virtual Machine” window in the virtual machine. This window will show the number of CPU cores available in the virtual machine. The virtual machine software will also show other information such as the amount of RAM and storage available. You can also view the type of processor and the speed of the processor in the virtual [machine software window](https://www.alibabacloud.com/product/machine-learning/pricing).

**2.3.8 Checking CPU Cores in the**[Command](https://www.alibabacloud.com/product/cloud-shell)**Line**

The [command line](https://www.alibabacloud.com/product/cloud-shell) is another way to check the number of CPU cores in a virtual machine. The command line can be accessed by entering the “Command Prompt” window in the virtual machine. This window will show the number of CPU cores available in the virtual machine. The command line will also show other information such as the amount of RAM and storage available. You can also view the type of processor and the speed of the processor in the [command line window](https://www.alibabacloud.com/product/cloud-shell).

**2.4. Examine memory in Virtual machine:**

Each virtual machine consumes memory based on its configured size, plus additional overhead memory for virtualization.

The configured size is the amount of memory that is presented to the guest operating system. This is different from the amount of physical RAM that is allocated to the virtual machine. The latter depends on the resource settings (shares, reservation, limit) and the level of memory pressure on the host.

For example, consider a virtual machine with a configured size of 1GB. When the guest operating system boots, it detects that it is running on a dedicated machine with 1GB of physical memory. In some cases, the virtual machine might be allocated the full 1GB. In other cases, it might receive a smaller allocation. Regardless of the actual allocation, the guest operating system continues to behave as though it is running on a dedicated machine with 1GB of physical memory.

**Shares**

Specify the relative priority for a virtual machine if more than the reservation is available.

**Reservation**

Is a guaranteed lower bound on the amount of physical RAM that the host reserves for the virtual machine, even when memory is over committed. Set the reservation to a level that ensures the virtual machine has sufficient memory to run efficiently, without excessive paging.

After a virtual machine consumes all of the memory within its reservation, it is allowed to retain that amount of memory and this memory is not reclaimed, even if the virtual machine becomes idle. Some guest operating systems (for example, Linux) might not access all of the configured memory immediately after booting. Until the virtual machines consumes all of the memory within its reservation, VMkernel can allocate any unused portion of its reservation to other virtual machines. However, after the guest’s workload increases and the virtual machine consumes its full reservation, it is allowed to keep this memory.

**Limit**

Is an upper bound on the amount of physical RAM that the host can allocate to the virtual machine. The virtual machine’s memory allocation is also implicitly limited by its configured size.

**2.5. Examine network resource in Virtual machine:**

**2.5.1. Network Virtualization and Virtual Networks**

**Network virtualization** is the process of combining hardware network resources and software network resources into a single administrative unit. The goal of network virtualization is to provide systems and users with efficient, controlled, and secure sharing of the networking resources.

The end product of network virtualization is the **virtual network**. Virtual networks are classified into two broad types, external and internal. **External virtual networks** consist of several local networks that are administered by software as a single entity. The building blocks of classic external virtual networks are switch hardware and VLAN software technology. Examples of external virtual networks include large corporate networks and data centers.

An **internal virtual network** consists of one system using virtual machines or zones that are configured over at least one pseudo-network interface. These containers can communicate with each other as though on the same local network, providing a virtual network on a single host. The building blocks of the virtual network are **virtual network interface cards or virtual NICs (VNICs)** and virtual switches. Solaris network virtualization provides the internal virtual network solution.

## 2.5.2. Parts of the Internal Virtual Network

An internal virtual network built on the Solaris OS contains the following parts:

* At least one network interface card, or NIC.
* A virtual NIC, or VNIC, which is configured on top of the network interface
* A virtual switch, which is configured at the same time as the first VNIC on the interface.
* A container, such as a zone or virtual machine , which is configured on top of the VNIC.

The next figure shows these parts and how they fit together on a single system.

##### VNIC Configuration for a Single Interface



 Figure -2.1

The figure shows a single system with one NIC. The NIC is configured with three VNICs. Each VNIC supports a single zone. Therefore, Zone 1, Zone 2, and Zone 3 are configured over VNIC 1, VNIC 2, and VNIC 3, respectfully. The three VNICs are virtually connected to one virtual switch. This switch provides the connection between the VNICs and the physical NIC upon which the VNICs are built. The physical interface provides the system with its external network connection.

Alternatively, you can create a virtual network based on the etherstub. Etherstubs are purely software and do not require a network interface as the basis for the virtual network.

A **VNIC** is a virtual network device with the same data-link interface as a physical interface. You configure VNICs on top of a physical interface. For the current list of physical interfaces that support VNICs, refer to the Network Virtualization and Resource Control FAQ. You can configure up to 900 VNICs on a single physical interface. When VNICs are configured, they behave like physical NICs. In addition, the system's resources treat VNICs as if they were physical NICs.

Each VNIC is implicitly connected to a **virtual switch** that corresponds to the physical interface. The virtual switch provides the same connectivity between VNICs on a virtual network that switch hardware provides for the systems connected to a switch's ports.

In accordance with Ethernet design, if a switch port receives an outgoing packet from the host connected to that port, that packet cannot go to a destination on the same port. This design is a drawback for systems that are configured with zones or virtual machines. Without network virtualization, outgoing packets from a virtual machine or a zone with an exclusive stack cannot be passed to another virtual machine or zone on the same system. The outgoing packets go through a switch port out onto the external network. The incoming packets cannot reach their destination zone or virtual machine because the packets cannot return through the same port as they were sent. Therefore, when virtual machines and zones on the same system need to communicate, a data path between the containers must open on the local machine. Virtual switches provide these containers with the method to pass packets.

## 2.5.3. Allocating Resource Control and Bandwidth Management on a Network

The following figure shows a corporate network topology that uses resource control to manage various applications.

Network With Resource Controls in Place


Figure -2.2

This figure shows a typical network topology that uses resource controls to improve network efficiency and performance. The network does not implement VNICs and containers, such as exclusive zones and virtual machines. However, VNICs and containers could be used on this network for consolidation and other purposes.

The network is divided into four tiers:

* **Tier 0** is the demilitarized zone (DMZ). This is a small local network that controls access to and from the outside world. Resource control is not used on the systems of the DMZ.
* **Tier 1** is the web tier and includes two systems. The first system is a proxy server that does filtering. This server has two interfaces, bge0 and bge1. The bge0 link connects the proxy server to the DMZ on Tier 0. The bge0 link also connects the proxy server to the second system, the web server. The http and https services share the bandwidth of the web server with other standard applications. Due to the size and critical nature of web servers, shares of http and https require guarantees and prioritization.
* **Tier 2** is the applications tier and also includes two systems. The second interface of the proxy server, bge1, provides the connection between the web tier and the applications tier. Through a switch, an applications server connects to bge1 on the proxy server. The applications server requires resource control to manage the shares of bandwidth given to the various applications that are run. Critical applications that need a lot of bandwidth must be given higher guarantees and priorities than smaller, or less critical applications.
* **Tier 3** is the database tier. The two systems on this tier connect through a switch to the proxy server's bge1 interface. The first system, a database server, needs to issue guarantees and to prioritize the various processes involved in database lookups. The second system is a backup server for the network. This system must consume a great deal of bandwidth during backups. However, backup activities are typically carried out overnight. Using resource controls, you can control when the backup processes have the highest bandwidth guarantees and highest priorities.

## 2.5.4. Who Should Implement Resource Control Features

Any system administrator who wants to improve a system's efficiency and performance should consider implementing the resource control features. Consolidators can delegate bandwidth shares in combination with VNICs to help balance the load of large servers. Server administrators can use share allocation features to implement SLA's, such as those offered by ASPs. Traditional system administrators can use the bandwidth management features to isolate and prioritize certain applications. Finally, share allocation makes it easy for you to observe bandwidth usage by individual consumers.

# 2.5.5. Observability Features for Network Virtualization and Resource Control

Network virtualization and resource control includes observability features to help you view resource usage before setting up controls such as VNICs and flows. In tandem with Solaris extended accounting, the resource control observability features allow you to accumulate systems statistics into logs. The observability features of network virtualization and resource control include:

* Ability to monitor a running system.
* Ability to log and report statistics.
* Extended accounting features to log historical data

The new flowed command and extensions to the dladm and netstat commands implement the network virtualization observability features. You can use these commands to monitor current system usage and to gather statistical data into logs.

# 2.6. Examining Virtual Machine Storage:

Virtualization is a wonderful technology. With it, I’m able to deploy and manage servers with a minimal amount of effort, which is absolutely great for today’s busy SysAdmin. Not to mention the other benefits you get, such as servers which can be scaled effortlessly, high availability and even power savings via server consolidation, to name just a few.

One important (and hopefully obvious) consideration when setting up new virtual machines, or modifying existing ones, is storage. Specifically, how is this handled and what are the best practises to follow? In this article, I’ll be looking at how this is done in a VMware environment and will try to offer a few solid guidelines to follow when setting up your virtualised disks and storage.

If you are unsure of anything that I’m covering (for example, some of the settings I recommend), it may be a good idea to backup any virtual machines or data you will be working with before starting. These processes become second nature once you have familiarised yourself with them, and soon you’ll be able to happily provision and manage virtual disks without worry.

## 2.6.1. Storage provisioning

To start with, we’ll assume that your backend storage is ready to go. You probably have a nice iSCSI or Fibre Channel SAN running alongside your host machines (the servers that will be running the virtual machines).

When you create a new virtual machine, you set its basic configuration options, one of which is the size of its virtual disk or disks. As a best practise, I tend to start by assigning the minimal amount of disk space needed for each virtual disk. After all, you don’t want to unnecessarily provision a truck load of free space that may not end up being used and, this way, you can make good use of all the disk space your current SAN provides you with. The same theory also applies if you are provisioning your virtual machines on a physical host’s local storage or a NAS device: Keep the sizes to a minimum and then expand as needed. This recommendation obviously changes if you are using thin provisioned disks, but I prefer to use “thick provisioned” disks whenever possible, as they seem to offer the best performance. (*In case you were wondering, thick provisioned is when you assign all of the specified disk space to your virtual disk at the time it is created.*)

As an example, let’s say we’re setting up a Windows 2003 Standard Virtual Machine as a Primary Domain Controller. This shouldn’t need too much disk space to be allocated at the start, so I would give it one virtual disk of about 6 or 7 GB to begin with. If you found that Windows was eating up disk space over time (don’t you just hate the rubbish that Windows update leaves behind?), your free disk space had dropped to below 10%, and you had cleared up as much temporary flotsam as you could, then it would probably be a good time to extend the size of the Virtual Machines’ disk.

So, let’s take a look at a scenario where you need to extend the size of a virtual machine’s disk. First of all you need to determine what is stored on this disk in question – is it a System disk? And by that I mean “is it the disk running Windows or the disk that contains the Paging file?” If so, this VM will need to be shut down in the case of older Operating Systems (see below for Windows Server 2008), and will require a little more work. If it’s just a data disk used for storage, then you’ll probably be OK to extend it on the fly – even while the machine is still running.

I always make sure to keep a special VM used just for extending virtual disks in our VMware clusters. This VM has a basic install of Windows 2003 Standard, and has the “**diskpart**” DOS utility installed by default. In terms of Operating Systems, most recent versions of Windows will do, (The key being NTFS partition support of course). Windows 2003 or 2008 would be recommended though, as diskpart has some small restrictions when it comes to extending partitions under Windows 2000 and XP. This machine is used to host an attached virtual disk from *another* VM that needs extending, though you should only need to use this VM to extend System and Paging disks on operating systems older than Windows 2008. This is because you shouldn’t power up a VM that has had its system or paging disk extended until you have *completed* the resizing operation, as it involves more than just extending the size in the VMware disk configuration. Instead, we need to attach the VM’s system or paging disk to this Windows 2003 VM as a secondary disk, and then boot into Windows and use diskpart to extend the size of the temporarily added disk. I’ll go into more detail below on how to complete the resizing operation for a System or Paging disk, and then later we’ll look at a Data disk. Note that Windows Server 2008 and above allow you to resize System or Paging disks whilst they are still powered on. For this OS, follow the VMware disk extending method below, and then refer to the Windows Server 2008 section for instructions to extend the disk “on-the-fly” in the OS itself.

*It’s worth bearing in mind that there are software tools such as Partition Magic and Partition Manager which can streamline this process, but these are paid-for tools, and the approach I’m describing here is possible using just the tools which you already have.*

## 2.6.2. System / Paging disks method

First and foremost, ensure that there are no snapshots on this VM. If there *are*, then you do not want to extend any disks, as this will result in data corruption. Commit or remove the snapshots before proceeding (See Figure 3), and once the snapshots are out of the way we can start by opening **diskmgmt.msc** (Windows Disk Management). Right click on the disk that requires extending (the grey area that reads: Disk X, XXGB, Online in Figure 4, for example), click **Properties** and note down the Location details. This will help you ensure that you have the correct disk when extending it in VMware, which you’ll do in a moment. An example location would be: *Location:  Bus Number 0, Target ID 2, LUN 0*.  See Figure 4 for more detail.

Now we need to shutdown the VM; wait for the machine to power off, and then go to the VM configuration (Right-click the VM, and select **Edit Settings**).  Select the disk you want to extend, and confirm it is the correct disk by matching up the location information we noted down from **diskmgmt.msc**. For example, a disk with *Bus Number 0, Target ID 2, LUN 0* as the information in Diskmgmt.msc would match up with the disk listed as *SCSI (0:2) Hard disk 2* in the VM’s “**Virtual Device Node**” information – Note that you are matching the Target ID of 2 with the SCSI (x:2) virtual device node information on the VM. You’ll obviously want to take care to make sure that these targets match up and that the current size of the disk looks correct!

Now just bump the disk size up to what you need, click **OK** (see Figure 5), and wait for VMware to complete the resize operation. Now, this is very important: do *not* power up yet! Leave this VM powered down and ensure it is not powered up until we are completely finished. If you don’t have a good memory for long strings of seemingly random information, it may be a good idea to note down this virtual disk’s location, as you’re going to need that information in a moment.

Go to the Disk managing VM that we mentioned earlier – the one with a basic Windows 2003 installation and **diskpart**– this VM should now be shutdown. Edit the settings of the VM, and under the “**Hardware**” tab, click the **Add** button. Select “**Hard disk**” as the type, click **Next,** and now we want to select “**Use an existing virtual disk**” (See Figure 6.) and browse to the exact VMDK file location of the Virtual disk we just extended. Be 100% sure that you have the right VMDK file, and then complete the wizard to finish attaching the disk to this VM. Wait for the add hardware task to complete, and then power up the ‘special’ VM.

Remember that if you are extending a System or Paging disk on Windows Server 2008 and above, you can skip the part above about ensuring the VM is shutdown first and the additional part about attaching the disk to your “disk manager” VM. You should still ensure you match up the disks though – you wouldn’t want to extend the wrong disk!

At this point there are two methods we can use to finish extending Windows Server 2003 or 2008 VM System or Paging disks. For  Windows Server 2003 disks, follow the **diskpart** instructions below. The Windows Server 2008 method will follow on below this.

## 2.6.3. Windows Server 2003:

Now that Windows has loaded, you may want to ensure that the disk we just added has been detected in **diskmgmt.msc**. Once you are happy that it can be seen here, fire up the command prompt, type in “**diskpart**“, and press enter. If you’re feeling efficient, you could even set this up as a startup item so that it loads as soon as Windows logs in. Next, follow this sequence (which you can see demonstrated in Figures 7 & 8):

* At the diskpart command line, type “**list vol**” and press enter; this lists the various disk volumes available to this VM.
* Next, type “**select vol x**” (where “x” is the volume number you want to extend) and press enter.
* Now type “**extend**” and press enter.  You should get a successful message.
* Type “**list vol**” again, just to ensure that the volume shows up as being the new extended size, and then type “**exit**” & hit enter to quit diskpart.

Shutdown your VM and be sure to remove your newly extended virtual disk. Make sure when you remove the disk that you only remove *the reference* to the disk from the VM; don’t remove the reference and delete your disk!  You may now power up the original VM again and check that the System or Paging disk has been extended as expected.

**2.6.4. Windows Server 2008:**

* As mentioned previously, we don’t need to shutdown these VMs before extending their System or Paging disks. Therefore there is no need for a “disk manager” VM to attach the disks to. Simply follow the process above to extend the System or Paging disk, omitting the part about shutting the VM down whilst extending the disk and attaching the extended disk to the “disk manager” VM.
* Once the VMware disk extension operation is complete, open **diskmgmt.msc**; You should see the disk you have added space to, with the additional space being displayed as “**unallocated**“. If you don’t see the unallocated space, refresh diskmgmt and it should then show up.
* Right-click the currently-allocated partition area of this disk and you should see the “**Extend Volume**” option. Choose this.
* Follow the “**Extend**” wizard to complete the resizing operation on this disk. It will ask you how much of the additional space you would like to use to extend the disk – usually you would choose to use all of it, but the choice is ultimately up to you.

Data disk method

Data disks are far easier to extend. As before, the first thing to do is to make sure that there are no snapshots on the VM in question. Remember: if you extend a disk with a snapshot still on the VM you *will* corrupt your data.

As a best practise, I usually ensure that there are no backups happening on the VM – i.e. no file level or server level backups are running, as I generally prefer to try and avoid doing disk operations on storage that currently has high IO. There is no concrete evidence that I am aware of to suggest that this theory is correct, but I like to stay on the safe side. Perhaps it has something to do with my thinking that a disk operation could potentially cause a short interruption to the current disk activity. If you wish to check the current activity, you can use the VM’s Performance monitoring tab to view current disk I/O or run “**perfmon**” in the Windows Operating system of the VM to measure disk activity.

Now that you are happy to proceed, check the disk’s Bus Number and Target id (e.g. *Bus Number 0, Target ID 1, LUN 0*) in **diskmgmt.msc**. For data disks you can leave the VM running as you extend them, so you can right-click the VM and choose “**Edit Settings**” right away. Select the correct hard disk by matching the Bus number & Target id information from diskmgmt.msc with the Virtual Device Node information you see on the VM; “*SCSI (0:1) Hard disk 2*” would be a match for the example information given above. Increase the size of this virtual disk, click **OK**, and wait for the operation to complete.

Next, you’ll need to go through a similar extension process as with the System disks:

* Run **diskpart.exe** from the command prompt and type “**list vol**” to display the disk volumes.
* Type “**select vol x**” where “**x**” is the volume number that you are extending.
* Finally type “**extend**” to extend the volume. You can double check the size has increased by doing another list vol at this point.
* Exit **diskpart** and you should be good to go with your newly extended data disk, no restart required!

A note on diskpart’s compatibility with certain disks, as quoted from Microsoft KB 325590:

*“In Windows XP and in Windows 2000, you cannot use Diskpart.exe to extend a simple volume on a Dynamic disk that was originally created on a Basic disk. You can extend only simple volumes that were created after the disk was upgraded to Dynamic disk. If you try to extend a simple volume on a Dynamic disk that was originally created on a Basic disk, you receive the following error message. This restriction was removed in Windows Server 2003.”*

Error message: *“Diskpart failed to extend the volume. Please make sure the volume is valid for extending”*.

## 2.6.5. New Virtual disks and alignment

Now that we’ve covered how to extend Virtual disks, this section will discuss creating *new* Data disks for Windows Guest Operating System Virtual Machines.  It’s worth noting that aligning the boot disk in the VM is, to quote VMware, “*neither recommended nor required*“. You only need to align *data* disks in your Virtual Machines.

Note that aligning the storage track boundaries is not required on Windows Server 2008 and above, as partitions created on these operating systems are aligned on 1MB boundaries. See the further reading links at the end of this article for more information. If you would like to learn more about storage track boundary alignment, these links also provide some very useful insight as to what it is and how it works (the short answer is that it’s a way of improving disk performance).

Add the new virtual disk to your VM, which can be powered on or off for this operation. Next, to ensure we get the best performance out of the virtual disk, we are going to align the storage track boundaries using **diskpart** on the new virtual disk.

* Once Windows has loaded, start **diskpart** from the command line and type “**list disk**” to get a list of your virtual disks.
* Select the correct disk with “**select disk x**” where “**x**” is the number of the new (and empty) virtual disk. Now we will create the partition.
* Type “**create partition primary align=64**” then press enter. Note you could also use 32 instead of 64 – it is best to adjust this depending on the recommendation from your storage vendor. If you are unsure, use 64.
* Now type “**Assign Letter=<DriveLetter>**” <DriveLetter> refers to the letter you would like to assign to this disk. For example “**Assign Letter=Z**” would assign your new disk as the Z: drive. Naturally, you need to make sure this is free before assigning.
* Now that we are done here, exit **diskpart**.

Now that you’ve created this partition, you can use the Disk Management snap-in or the Windows Format command to format it as an NTFS partition. For Windows Guest Operating systems, VMware recommends formatting the data disk with an allocation unit size of 32K.

**2.6.6. Virtual machines: virtual computers within computers**

A virtual machine, commonly shortened to just VM, is no different than any other physical computer like a laptop, smart phone, or server. It has a CPU, memory, disks to store your files, and can connect to the internet if needed. While the parts that make up your computer (called hardware) are physical and tangible, VMs are often thought of as virtual computers or software-defined computers within physical servers, existing only as code.



## Figure -2.3

## 2.6.7. How does a virtual machine work?

Virtualization is the process of creating a software-based, or "virtual" version of a computer, with dedicated amounts of CPU, memory, and storage that are "borrowed" from a physical host computer—such as your personal computer— and/or a remote server—such as a server in a cloud provider's datacenter. A virtual machine is a computer file, typically called an image, that behaves like an actual computer. It can run in a window as a separate computing environment, often to run a different operating system—or even to function as the user's entire computer experience—as is common on many people's work computers. The virtual machine is partitioned from the rest of the system, meaning that the software inside a VM can't interfere with the host computer's primary operating system.

**2.7. Understanding Virtual machine clones:**

**2.7.1. What is VM cloning?**

A VM clone is a copy of a virtual machine. The existing virtual machine is known as the parent, while the new VM is called the clone. After the cloning operation, the clone VM runs as a separate virtual machine.

Why would you need to clone a virtual machine? Cloning is a fast and simple way to create a new virtual machine that shares properties with an existing one. The process of installing a guest operating system and programs from scratch can take a great deal of time. Using cloning, you can perform installation and configuration once, and then use the clone as a basis for many future virtual machines.

VM cloning is most useful for deploying multiple identical virtual machines to a group of users. For example, a sysadmin can clone a virtual machine for each employee in a particular department—since the employees use the same applications, their setups should be the same. Similarly, a teacher may wish to clone a virtual machine for each student, with lesson materials and programs preinstalled. VM cloning is also helpful for software testing. Testers can clone a development environment and use it as a baseline for comparison while testing.

There are two types of VM clones: full clones and linked clones. A full clone is a completely separate copy of a VM that shares no system resources with the parent once it’s running. A linked clone, on the other hand, continues to share virtual disks with the parent after it’s created. Since it runs independently, a full clone generally has faster performance than a linked one. However, a full clone can take longer to create, with delay times of up to a few minutes when file sizes are large. Linked clones are faster to create and have the advantages of saving disk space by allowing multiple VMs to run off a single software installation.

**2.7.2. What is the difference between a clone and a snapshot in VMware?**

VMware allows users to create either a clone or a snapshot of a virtual machine. A snapshot preserves the current state of a virtual machine, copying the VM’s disk file. It is used for backup purposes. If you need to save the configuration of a virtual machine so you can revert back to it later if something goes awry, create a snapshot. A clone, by contrast, is an entirely separate copy of the VM. If you need to make a new VM that will run independently, cloning is the only way to go.

# 2.8. Virtual Machine Template

Templates are primary copies of virtual machines that you can use to create virtual machines that are ready for use. You can change the template, such as installing additional software in the guest operating system, while preserving the state of the original template.

When you clone a virtual machine from the vCenter Server inventory to the content library, you can choose what type of content library item to create. You can choose to create a library item of either the VM Template type or OVF Template type.

## Procedure

1. Navigate to the virtual machine or template that you want to clone.
2. Select your task.

| **Option** | **Description** |
| --- | --- |
| **Clone a virtual machine** | * 1. Right-click the virtual machine and select **Clone** > **Clone as Template in Library**.

The **Clone Virtual Machine To Template** wizard opens.* 1. On the **Basic information** page, enter a name and description for the template, select the template type, and select an inventory folder for the template.

You can create an OVF Template or VM Template in the content library.* 1. On the **Location** page, select a local content library in which you want to add the template.
	2. On the **Select a compute resource** page, select the compute resource for the template.
	3. On the **Select storage** page, select the storage for the template disk and configuration files.
	4. On the **Review** page, review the details and click **Finish** to complete the cloning task.
 |
| **Clone a virtual machine template** | * 1. Right-click the virtual machine template and select **Clone to Library**.

The **Clone to Template in Library** dialog box opens.* 1. Select the **Clone as** option.

You can create a template or you can choose an existing template to update.* 1. From the content libraries list, select the library in which you want to add the template.
	2. Enter a name and description for the template.
	3. **(Optional)**Select the configuration data that you want to include in the template.

You can select to preserve the MAC-addresses on the network adapters and include extra configuration.* 1. Click **OK**.
 |

## Table-2.1

## Results

A new task for cloning appears in the **Recent Tasks** pane. After the task is complete, the template appears in the **Templates** tab for the content library. You can view the type of template in the **Type** column.

## 2.9. Virtual Machine Snapshot

A virtual machine snapshot (VM snapshot) is the state of a virtual machine (VM) that is copied and stored at a specified time. It develops a copy of the VM that is used for VM migration, backup and restore procedures. A virtual machine snapshot allows a VM to be restored to a former state of snapshot creation.

A virtual machine snapshot is also known as a virtual machine image (VM image).

A virtual machine snapshot works as a typical operating system (OS) snapshot. Its primary purpose is to create an exact VM replica. A virtual machine snapshot is created by the client/server hypervisor or the VM manager.

The snapshot keeps the following records:

* **State**: Includes the operational state of the VM (such as active), which is suspended along with its configuration.
* **Data**: Includes all files from disk, memory and device driver cards.

A virtual machine snapshot is also important for operational environment, where the same instance of a VM must be created multiple times.

### 2.9.1. What is Open Virtualization Format (OVF)?

Open Virtualization Format (OVF) is an open source standard for packaging and distributing software applications and services for virtual machines (VMs).

As the adoption of virtual infrastructure increases, there is a greater need for an open, standard, portable and platform-independent metadata format to distribute virtual systems onto and between virtualization platforms. OVF provides such a packaging and distribution format to facilitate the mobility of VMs.

The standard also describes multiple VMs with their relationships. These VMs can be wrapped up in a single virtual appliance file to enable broader distribution.

### Open Virtualization Format explained

OVF is not a specification describing a virtual disk. Rather, it is a standard representation of VM metadata. This VM metadata includes the following:

* name
* configured memory
* CPU
* storage settings
* network

In addition to describing the above attributes of virtual hardware, OVF also allows virtual appliance vendors to add comments about the VM and other characteristics, such as an end-user license agreement (EULA), boot parameters and minimum requirements. They can also encrypt, compress and digitally sign their content.

OVF, which is specified by the Distributed Management Task Force (DMTF) and published by the International Organization for Standardization (ISO) as ISO 1720, is independent of any particular processor or hypervisor architecture. It leverages DMTF's Common Information Model (CIM) to allow management software to understand and map resource properties by using the OVF open standard.

As a packaging format for virtual appliances, OVF enables the mobility of virtual machines across multiple platforms by facilitating the distribution of enterprise software in a flexible, secure and efficient manner.

Consequently, both vendors and users can follow OVF specifications to deploy a VM on any virtualization platform. They can take full advantage of virtualization's benefits, including the following:

* enhanced flexibility
* portability
* verification
* version control
* signing
* better licensing terms

### Features of Open Virtualization Format

Key features of OVF are as follows:

**Validation support.**OVF supports the validation of every VM and the complete package.

**Supports single and multiple VM configurations.**OVF supports both single VM packages and complex multi-tier package services involving more than one interdependent VM.

**Content verification support.**Depending on the industry-standard public key infrastructure (PKI), OVF enables integrity checking and content verification.

**Licensing support.**OVF supports management and software licensing strategies.

**Platform-independent.** OVF was designed to be platform-independent, whether it's a guest OS, host platform or virtualization platform.

**Extensible.**OVF can support new technological advancements in virtualization and virtual appliances.

**Portable packaging.**OVF allows vendors to add platform-specific enhancements to their appliances and software.

### 2.9.2. Open Virtualization Format package

An OVF package is a group of files required to import the VM. These files are generally found in the same folder and they can be compressed, digitally signed, encrypted and archived. The package consists of metadata and file elements that describe the VMs, as well as additional information required to deploy and operate the applications in the package.

The metadata or descriptor file is one of the components of the OVF file. This extensible markup language (XML) document with the extension of .ovf includes details such as the location of virtual disks associated with the VMs, plus information about managing the VMs during import.

In addition, the OVF package typically includes the following files:

* The manifest file with a .mf extension that references the OVF files and their checksum.
* A certificate file with the .cert extension if a user chooses to digitally sign the OVF file.
* Files comprising virtual disks in the format specified by the virtualization product that exported the virtual disks.

An OVF package can be used by multiple stakeholders, including the following:

* An independent software vendor (ISV) publishing a software solution.
* A data center operator aiming to transport a software solution between data centers.
* A customer trying to archive software.

In general, any user or use case that can benefit from a standardized package for a software solution can use the OVF standard.



## Figure -2.4

Uploading OVF template in vSphere Client.

### Distribution of Open Virtualization Format packages

OVF packages can be distributed either as a single file or as a set of files.

#### Distribution as a single file

The OVF package is stored as a single file in the tar format. In OVF tar files, duplication is not allowed within the archive. The files appear in the following order inside the archive:

* .ovf descriptor
* .mf manifest (optional)
* .cert certificate (optional)

To find a file, a tar extraction tool scans the whole archive, even if the requested files are found in the beginning.

#### Distribution as a set of files

The OVF package can also be available as a set of files. This is common for applications involving a standard web browser.

The DMTF provides this example of an OVF package as a set of files:

http://xyzwebsite/virtualappliances/package.ovf
http://xyzwebsite/virtualappliances/virtualdisk1.vmdk
http://xyzwebsite/virtualappliances/virtualdisk2.vmdk
http://xyzwebsite/virtualappliances/additionalresource.iso



## Figure -2.5

Choosing a location and naming for the VM appliance created using the OVF template.

### 2.9.3. Open Virtual Appliance in an OVF File

An Open Virtual Appliance (OVA) is an OVF package in a single file archive. Its file extension is .ova. An OVA package, which is in the tar format, contains the files comprising an OVF package.

The OVA package is one large file that doesn't have the flexibility of the OVF package, which is a series of uncompressed files. This is because users can access the individual disk images in the files in the OVF package -- which they can't do with the OVA package. Another drawback of OVA packages is that it takes longer to export and import them.