STUDENT NAME: Adam Kenneth Shibaba

STUDENT ID: C00242437

PROJECT TITLE: Virtual Docker Testbed for Cybersecurity

SUPERVISOR: James Egan

SUBMISSION DATE: 26/04/2022

ABSTRACT

Most organizations are rapidly advancing towards implementing a system to support several online operations and this includes collaborative, remote, development and testing, among various jobs, and with such a change, IT specialists or organisations are consistently focused on managing their demands. An element that has contributed to this change is virtualisation. In a nutshell, virtualisation is a virtual environment that allows users perform several IT operations like a physical machine; however, it is not a real machine, but they exploit the resources of a physical device. In this documentation, I will explore virtualisation including the different tools that assist its technology, as well as a collective evaluation. With such information, I do not only intend to fulfil the development of the virtual environment as part of this project, but to provide the reader with a great level of understanding and factors to consider when deciding on a virtual tool to work with.

Contents

WHAT IS VIRTUALIZATION?

Virtualisation is a process that is backed up with software to develop a virtual environment for various goals relevant to IT operations such as storage, networking, development, design and testing.

Types of virtualizations

There are three major types of virtualizations, and they are Server Virtualization, Desktop Virtualization and Network Function Virtualization.
Server Virtualization: Consists of para virtualisation, full-virtualisation, and OS level-virtualisation

- A para virtualization functions in a hypervisor nature. Most software-based virtualization coexists with emulation overhead that utilizes this model. Para virtualisation will require a guest operating system to undergo modifications and recompiling before it is embedded into a virtual machine. Such changes improve the performance of the guest machine such as directly communicating with the hypervisor instead of the emulation overhead. For example, Xen (virtualisation tool like hypervisor) would utilize para virtualisation in a custom built Linux environment to configure an administrative environment called domain 0 (virtual machine).

- Full Virtualization: Uses a hypervisor software that communicates with a physical server to access resources such as CPU and disk space to be allocated to virtual servers running on the physical server. Also, full virtualization becomes the virtual servers' operating system, thereby making it possible for each virtual machine to operate independently without any awareness of the other.
  [1]

- OS-level: Does not utilize hypervisor because it becomes part of the host OS which performs tasks that are synonymous to a fully, structured, hypervisor virtual environment. A downside to this type is it does not permit guest servers to run different operating systems, thereby limiting them to the same OS and capabilities; however, each virtual machine remains independent from others and with such limitation, it only means virtual servers exist in what is called a homogenous environment.
  [2]

Para Virtualization Diagram

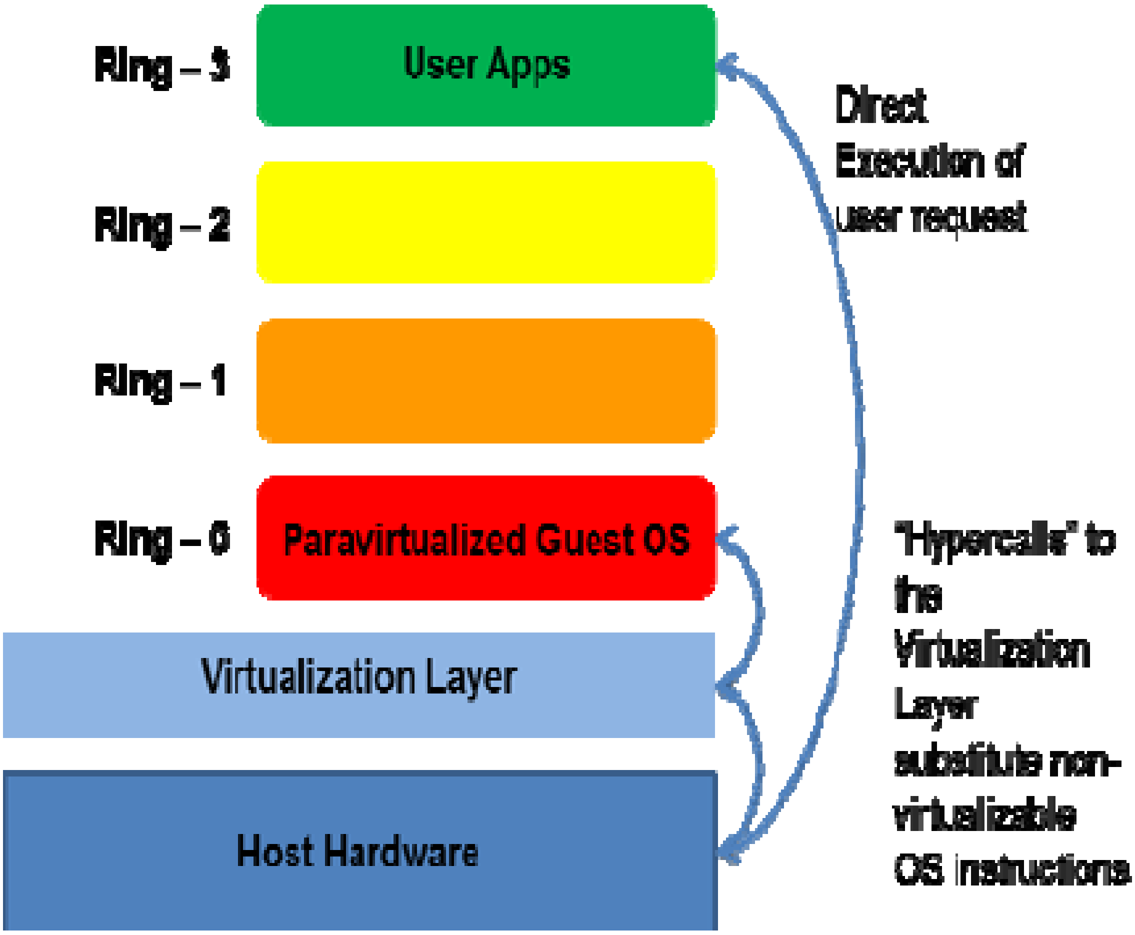

Ring – 3
User Apps
Ring – 2
Ring – 1
Ring – 0
Paravirtualized Guest OS
Virtualization Layer
Host Hardware
Direct Execution of user request
"Hypercalls" to the Virtualization Layer substitute non-virtualizable OS instructions

Diagram1.0
[3]

Full Virtualization

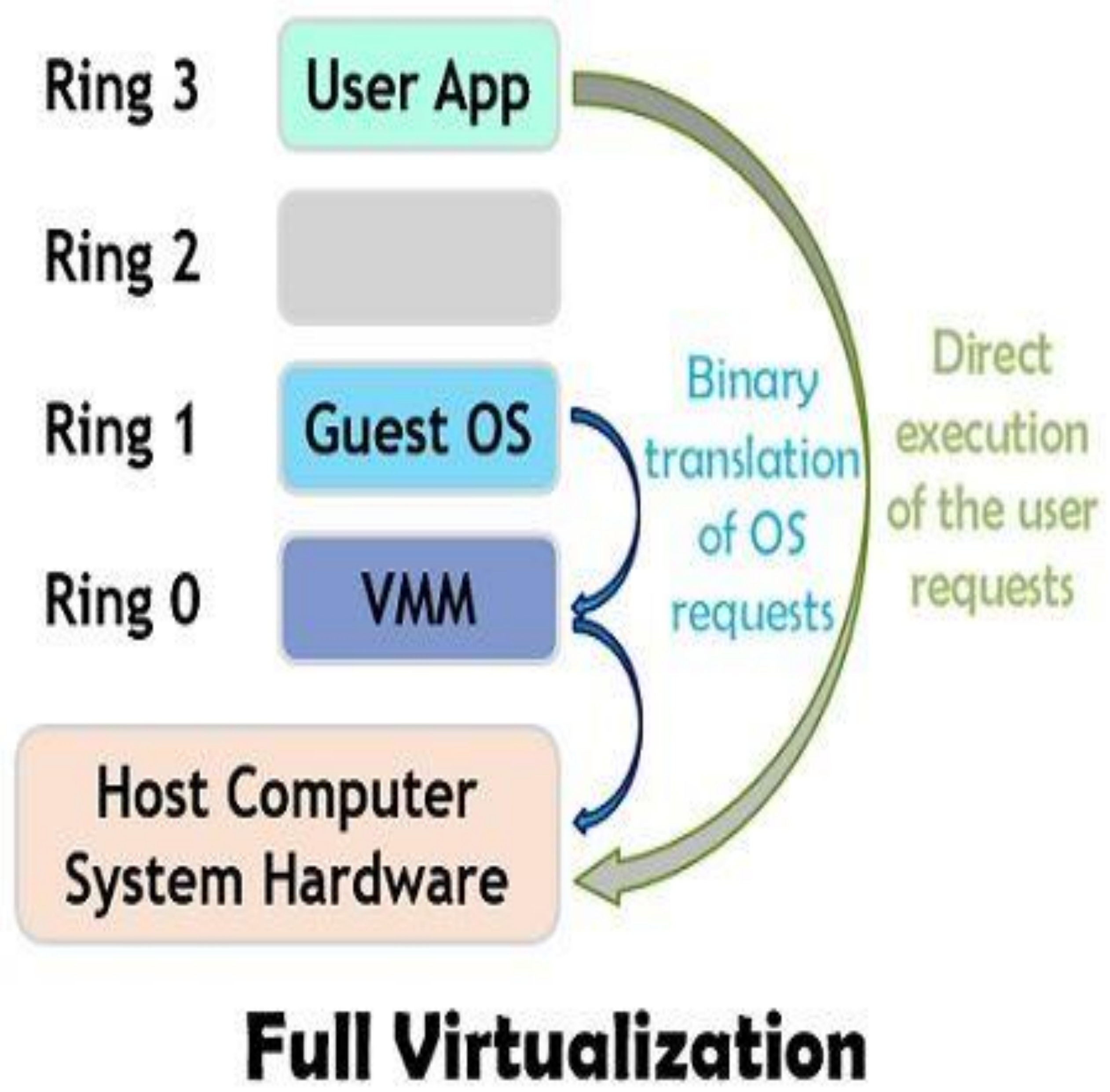

Ring 3
User App
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host Computer System Hardware
Binary translation of OS requests
Direct execution of the user requests
Full Virtualization

Diagram 1.1
[4]

OS Level Virtualization

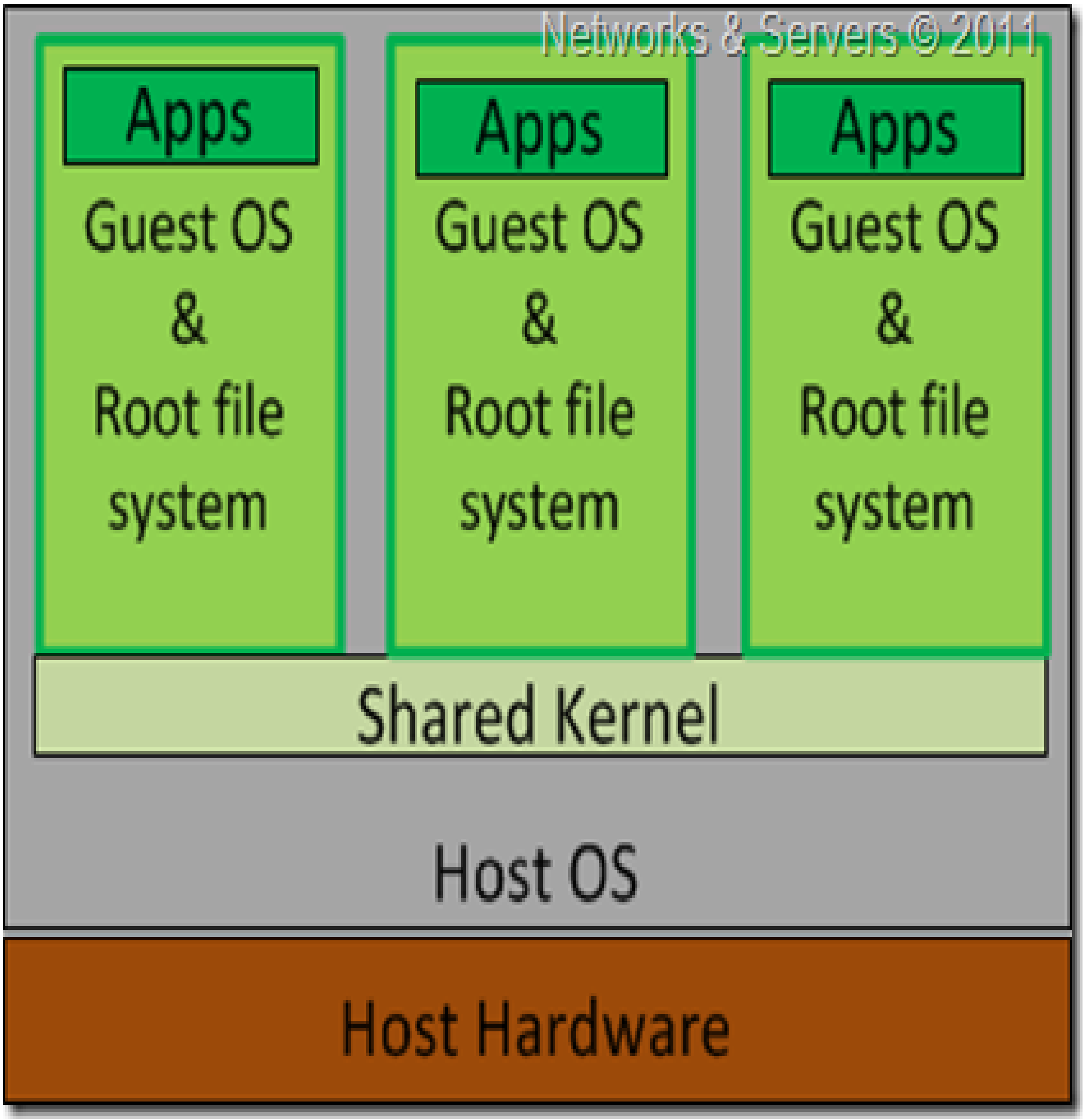

Networks & Servers © 2011
Apps
Guest OS & Root file system
Apps
Guest OS & Root file system
Apps
Guest OS & Root file system
Shared Kernel
Host OS
Host Hardware

Diagram 1.2
[5]

Desktop Virtualization: This type of virtualization allows the deployment of various operating systems in a single physical device which assists administrators in fulfilling tasks such as configuration, installation of extensions and packages, updates, and security checks for all virtual desktops. An example of desktop virtualization is Microsoft Remote Desktop Services which enables users/administrators to utilize specific applications to log into a server to access a virtual desktop for the purpose of managing or processing data or services. An application that is used for such operations is remmina which assist Linux users to remotely connect to Windows servers, using an RDP (Remote Desktop Protocol).

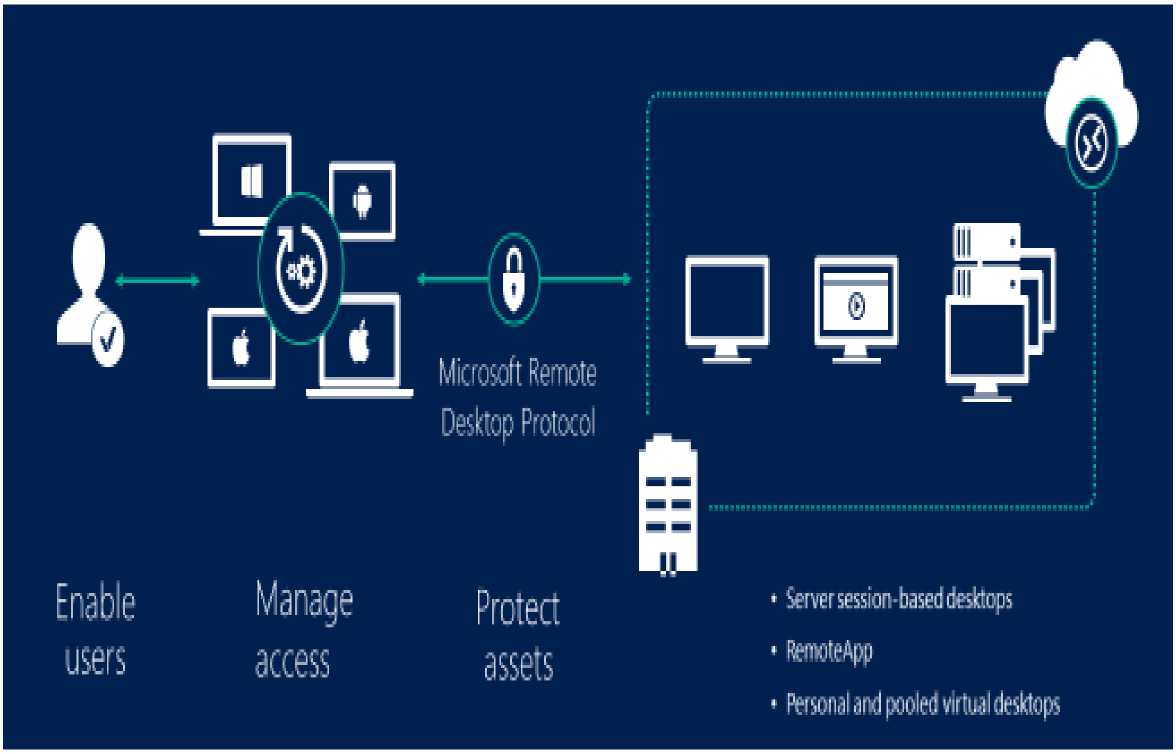

Microsoft Remote Desktop Protocol
Enable users
Manage access
Protect assets
• Server session-based desktops
• RemoteApp
• Personal and pooled virtual desktops

Diagram 1.3
[6]

Network Function Virtualization: The acronym NFV was created as an alternative to proprietary network software functions such as firewall, DNS, encryption, switches, and routers as virtual functions to assist organizations escape the cost-effective strains that standard proprietary software was causing them especially since such functions required specific hardware to perform. This way, companies can run their network on standard servers rather than proprietary ones.

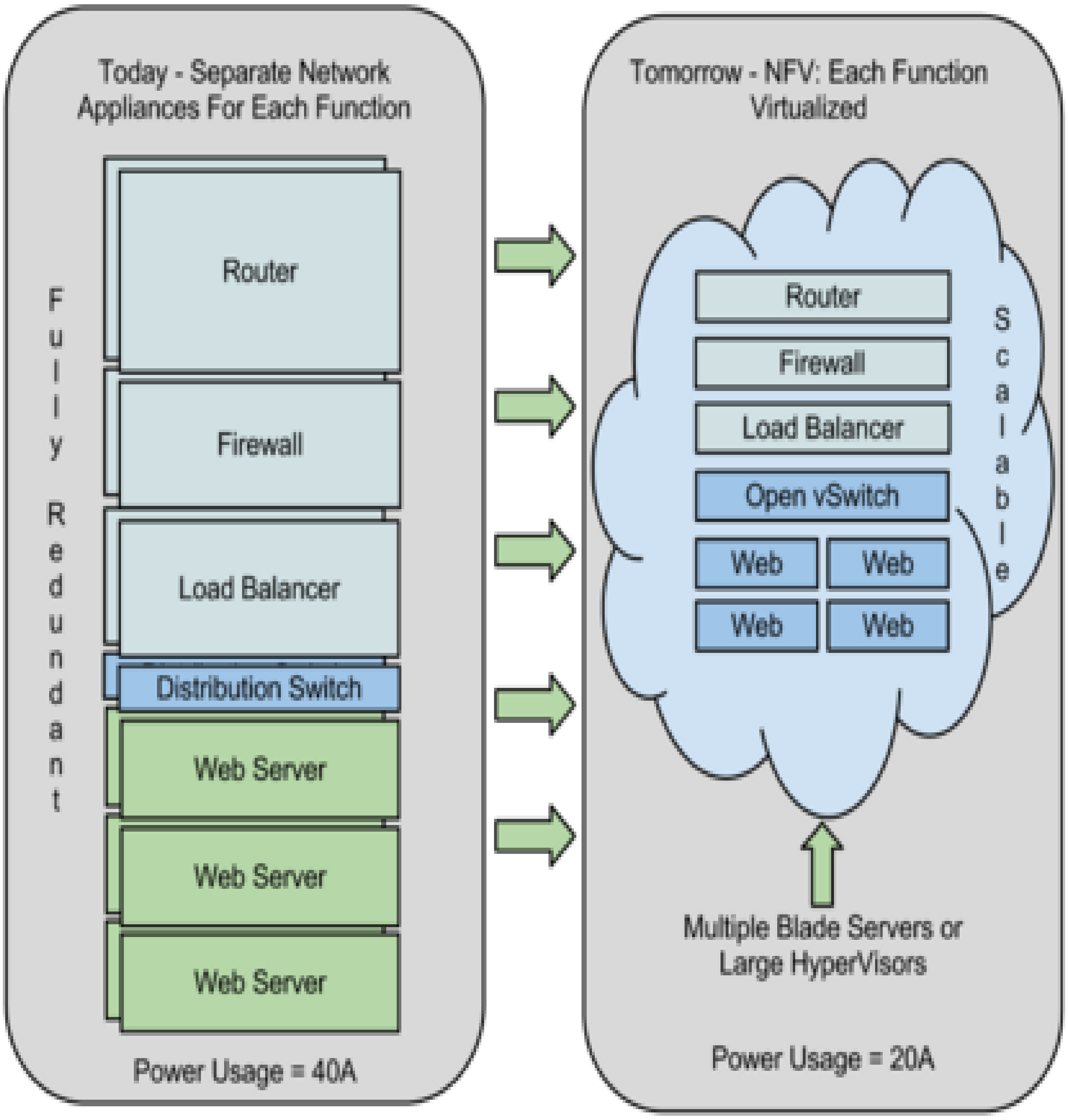


Diagram 1.4
[7]

HOW VIRTUALIZATION FUNCTIONS

Virtual implementation occurs with a software known as hypervisor. This tool incorporates the resources of a physical device (host) it operates within, to manage virtual machines (guests), installed in the hardware. The management of virtual machines will require hypervisor to utilize various internal components of the host such as CPU, RAM, and storage to be allocated among or between guests, including scheduling processes for them. Though this is an operation hypervisor is involved in, the physical hardware remains essential since it still handles the execution, therefore, any request the vms make such as CPU, instructions would be delivered by the component while the hypervisor software would administer the scheduling.

Hypervisors consist of two types.

- Hypervisor Type 1: Under this category, implementing a virtual base will require installing various famous hypervisor 1 tools such as Hyper-V, Esxi or KVM in a server that has ***no*** operating system and applications that are intended to be worked with by an organisation are included. For example, a company that has decided to use this type of Hyper-V for its virtual environment would only install this program into the server followed by the applications (installed and executed) it intends using such as Microsoft Exchange or IIS Services. This process would be completed without an operating system.
- Hypervisor Type 2: A type 2 hypervisor involves an additional layer in a virtual architecture. Contrary to a type 1, it requires the host to contain an operating system which it would be installed on, followed by selected virtual machines that would be executed known as guests, thereby using the potentials of inner components to facilitate the performance of the vms. A type 2 hypervisor is usually configured on devices such as desktops or laptops.

[8]

HYPERVISOR DIAGRAM

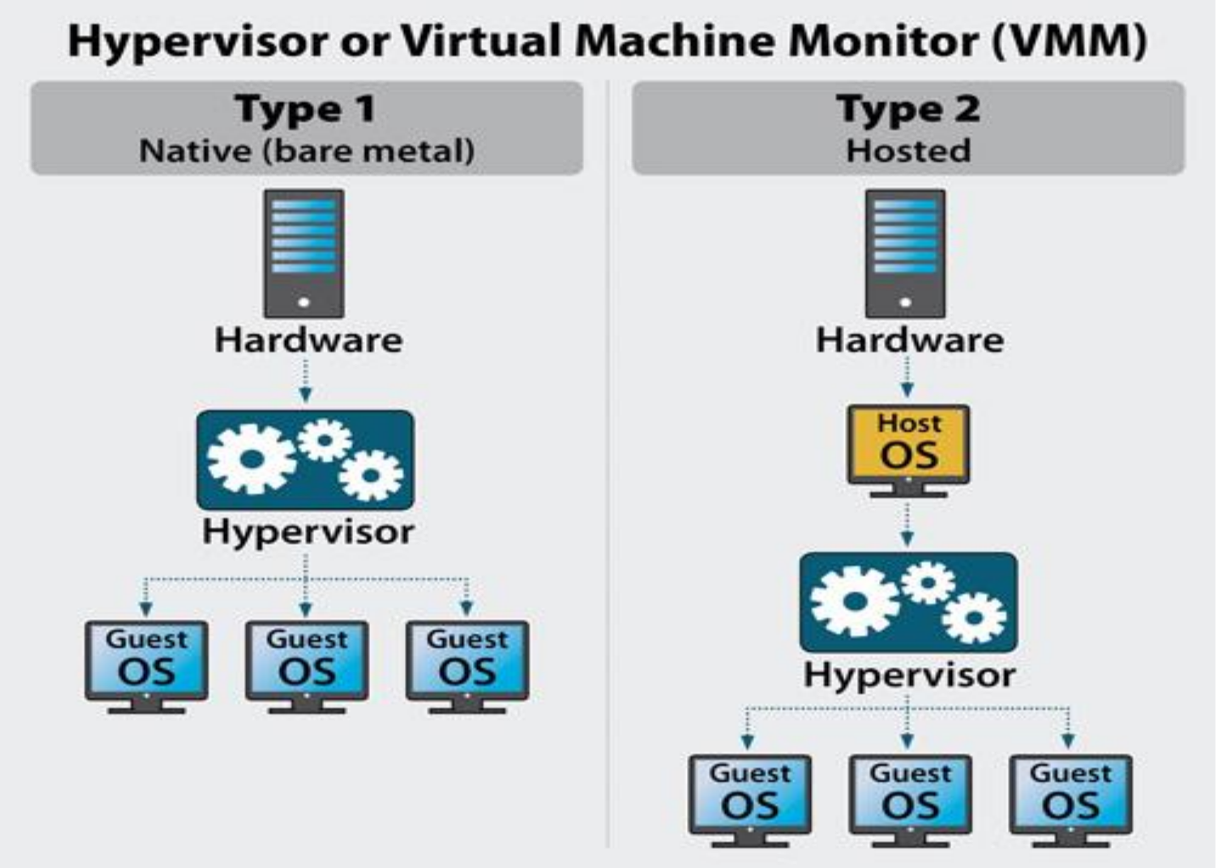


Diagram 2.0
Diagram above displays what is involved in the different types of hypervisors. As mentioned, the hyper-v 1 type does not require an operating system. The hypervisor software is stationed between the server and guest OS. So installation of a windows 2012 can be done without an operating system and executed.

A type 2 would be contrary to a type one as it is demonstrated in the diagram. This type would require the host operating system to be present to install applications or VMs that would be allocated resources and then executed.
[9]
[10]

Benefits of virtualization

- Virtualization is not cost effective: This is because it does not require hardware. What is required is the license or access from third party that maintains all servers.
- Helps to reduce expenses: Organizations spend extensively most times on computing operations because their infrastructure could be faced with changes to hardware and unknowingly, they end up possessing additional equipment. With virtualization, organizations can evaluate their systems and determine the hardware and resources they require for work which in return helps to avoid any waste.
- Helps with resiliency: Since deploying hardware can be time consuming perhaps by days, weeks or even months, virtualization helps organizations to handle deployment conveniently by creating measures to back up, copy, clone, and store VMs in different locations. In the event of an accident or disasters, virtualization would help companies continue with tier operations.
- Availability: Virtualization creates redundant virtual environments that ensure availability. A huge advantage not only with regards to security breach but to administrators when managing a virtual server remotely anywhere in the world. Also, availability can reduce the strain of downtime during operations.
- Easy Devops: Virtualization is able to facilitate devop tasks by breaking them into segments or test stages which involves cloning of VMs, to determine issues such as bugs, and with such method, devop engineers are able to complete the development process of a product without affecting it. This also means devop engineers are not faced with limitations associated with physical hardware because virtualization provides instant access to replicated VMs for developers to utilize efficiently.

[11]

Disadvantages of virtualization

- Implementation: It requires more investment to fulfil implementation. This is due to the fact the hardware and software required to expand IT operations especially with regards to devices to be utilized can be expensive however, such high investment can be deemed as a onetime purchase with long term good
- Limitations: There are not a lot of limitations in virtualization. This is because not all servers or applications possess compatible features for virtualization. If organizations need to utilize software that is not virtualization compatible, then it would have to be purchased, especially since most of their IT infrastructure might not be suitable for some virtualization components. Also, because some vendors do not provide support to assist organizations for compatibility, companies have begun migrating to what is known as Hybrid Systems.
- Security: Since data is a huge asset in organizations, data security in a cloud environment is crucial. Taking into account that a server is managed by the organization and third-party provider, then it is necessary to select an effective security measure wisely in order to maintain strong protection.

- Availability: As we know, data must be stored for a long period of time and if this cannot be maintained then an organization would be faced with losing its position in the competition race against its adversaries. Also, an issue around availability can occur from virtualization servers going offline due to network problems, affecting websites hosted by the servers, controlled by third parties which can result in organizations being in a helpless state.
- Scalability: This can be a huge challenge especially for SMEs that do not have the potentials to expand their business, giving fast growing organizations an upper hand in dominating small businesses and furthermore exploiting their resources.

[12]

Virtualization in cloud computing

In a simple way to explain it, cloud computing is a way of accomplishing several tasks in a virtual environment that exists over the internet. The same rules apply when creating our environment such as hypervisor, OS, networks, and VMs. The only difference is rather than allow our operations to be performed using the resources of a centralised system such as location or physical machine, we can complete tasks over the internet meaning we can store, access, and manage applications most importantly, access them at anytime from anywhere in the world.

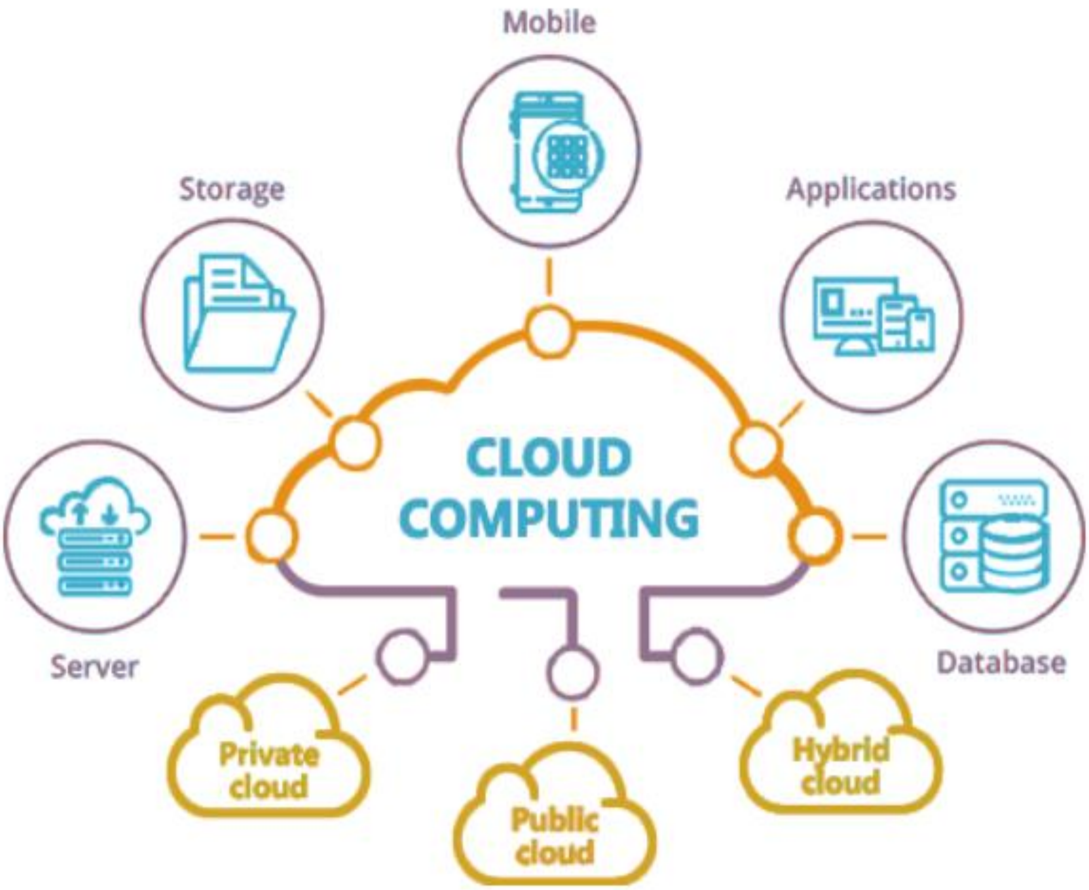


Diagram 2.1

Types of cloud computing

- Public: Are computing services supported by third party providers to make available for individuals or organizations and such services only occur over an internetwork (including intranet work). Some of these services are made free or involve finance, however, cloud providers have created a sense of convenience to make customers pay per usage for services or resources such as CPU cycle, storage, bandwidth and even applications. Public clouds help companies to save hugely on cost especially as the infrastructure that supports their operations are managed and maintained by the cloud provider(s) and this makes the benefit of cloud computing stress free. Some benefits that come with public cloud is they can be installed faster than on premises IT infrastructures including infinite scalability. Access to applications by employees can be achieved from any location if employees have access to a device and the internet. Security would always be an issue but if service providers apply proper security measures, then our cloud environment can be less prone to security issues. A good security implementation is intrusion detection and prevention systems (IDPS). [13]

- Private: Private relates to computing services that facilitate operations not only over the internet but a private internal network to satisfy the operations of specific users rather than the general public. Also known as corporate cloud, this system provides businesses several advantages, similar to a public cloud such as scalability, self-service and elasticity including control of resources over the infrastructure provided for them. In private cloud, security and privacy potentials are appreciable because of features such as firewalls and internal hosting which helps to maintain the safety of sensitive data and inaccessibility to third-party providers. Cloud computing consist of two models: **Iaas** which stands for infrastructure as a service provide for organizations such as network, compute, and storage as a service. The second model is **Paas** (platform as a service) which provides companies with application tools from cloud-based to enterprise-based
[14]
- Hybrid: Hybrid cloud is a cloud environment that consists of both private cloud which can be referred to as on-premises datacentre and public cloud, which allows operations involving data and applications to be exchanged between both systems. Some of its benefits include providing companies with scalability: this is effective when a company's processing requirements increases beyond an on-premises level. Also, it assists organizations to avoid cost for maintaining, installing and purchasing of they do not necessarily require. Hybrid security measures present very good potentials because they possess up to date, automated data redundancy, high availability, disaster recovery, and cybersecurity features.
[15]

- Multiclouds: Multi cloud is a cloud computing system consisting of 2 or more cloud platforms. It is important to understand this is not the same as a hybrid cloud. One reason this type of cloud stands out is because of one of the requirements it works

with which is Kubernetes, a powerful tool that facilitates the growth and modernization of legacy applications. Some of its benefits include availability on a wide range where if one cloud environment stops working, there is another cloud supporting applications. It provides better user experience because different users can be routed to the nearest cloud to ensure lower latency and better experience. Multi cloud can be a mechanism to enable specific integration especially for actions that work on a specific cloud. For example, if an organization has sensitive firewall data that it does not want to be stored in a public cloud but it needs to build integrations for private, multi cloud approach can allow the creation of workloads on the private side that can interact with such sensitive data.
[16]

Virtualization security

Virtualization is faced with security issues and in this area I would be highlighting on how some security matters can occur in a virtualized environment. Also, it is crucial the chances of their presence is always taken into serious consideration at every stage from design, configuration, installation and deployment.

Security Considerations in Virtualized Environments

VM SPrawl

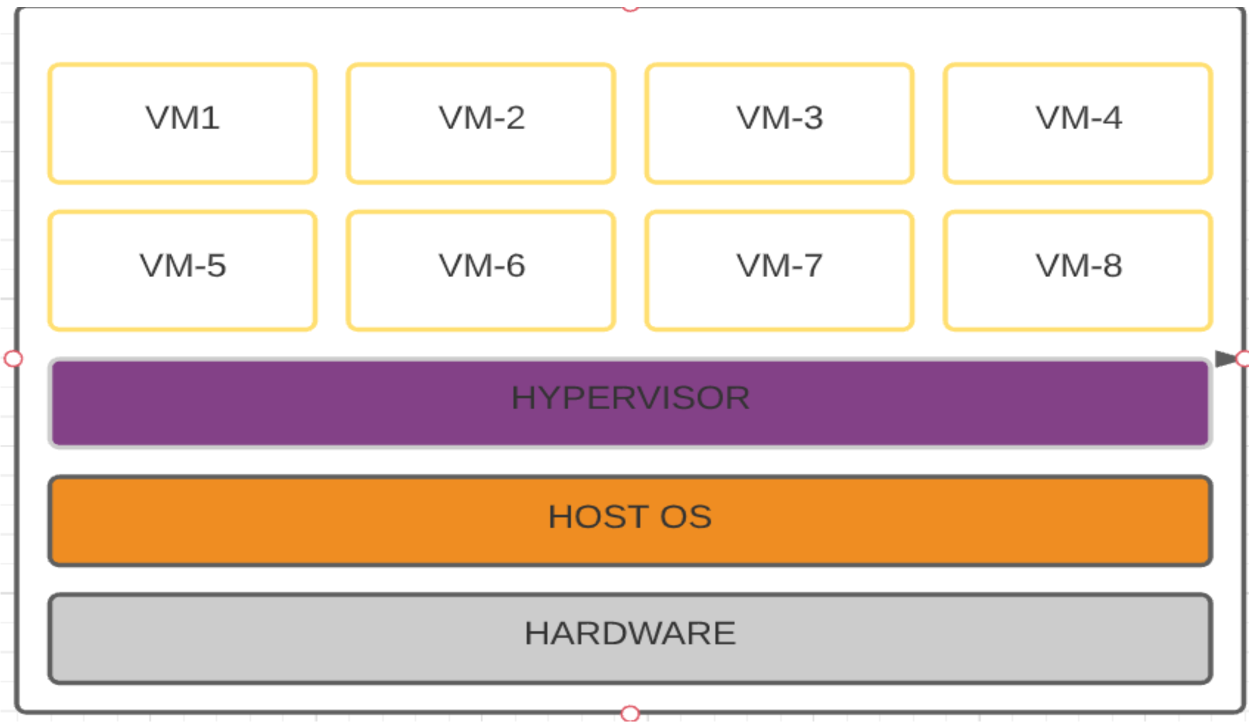


Diagram 3.0

IT specialists such as administrators create several virtual machines and sometimes, they are faced with the difficulty of managing these instances and as a consequence, their systems can possibly be prone to an attack due to the fact the administrator might not be able to keep full attention (manage) of VMs, especially as the virtualization environment continues to expand.

Let us look at diagram 3.0. In this image we have a virtual environment with few VMs. Imagine this environment begins to expand with more vms created and VM-1 contains an outdated and unpatched operating system that was forgotten about by the administrator, this can create an opening for an attack which could allow the attacker access other virtual machines and the data they contain. Also, not only can an organization suffer from the harm of an attack but, if VM sprawl is not managed well, this creates a problem of cost. As VMs are being created they can consume huge levels of resources belonging to a physical machine such as storage, CPU, RAM etc. Furthermore, the licensing of software and operating system these machines would require to work. These are problems that can be caused by VM Sprawl. So the lack of control due to VM sprawl can contribute to security issues in virtualization.

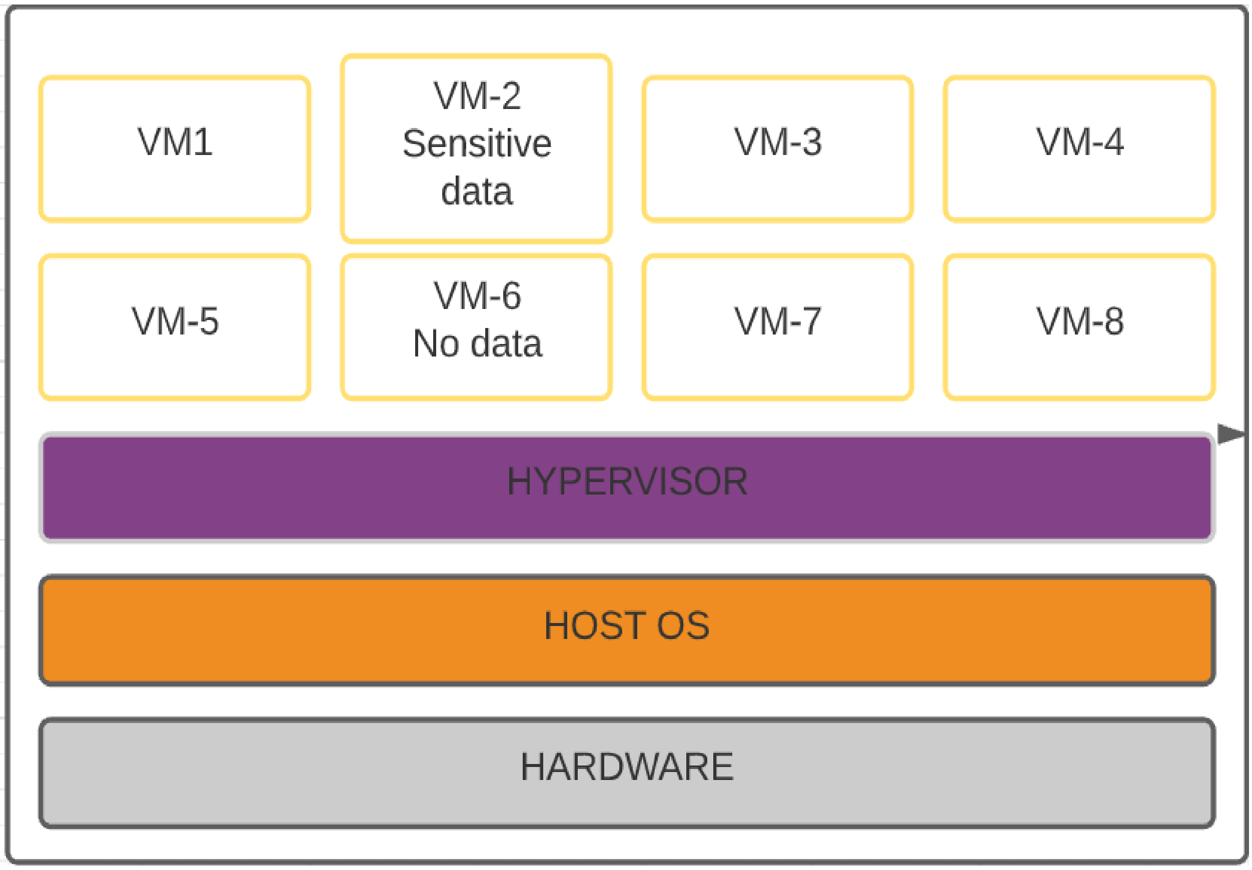


Diagram 3.1

Let’s assume VM-2 contains very sensitive data and VM-6 possesses nothing. This can be a potential security risk if these instances are utilizing the same physical machine because where there is a compromise with VM-6, such event could create an opening for the data in VM-2 to be explored and exploited. It is crucial for administrators take this into consideration when creating virtual environments including implementation of security policies for the vms they create. This means lack of separation could cause less visibility of what is happening in our virtual environment.

Blind Spot

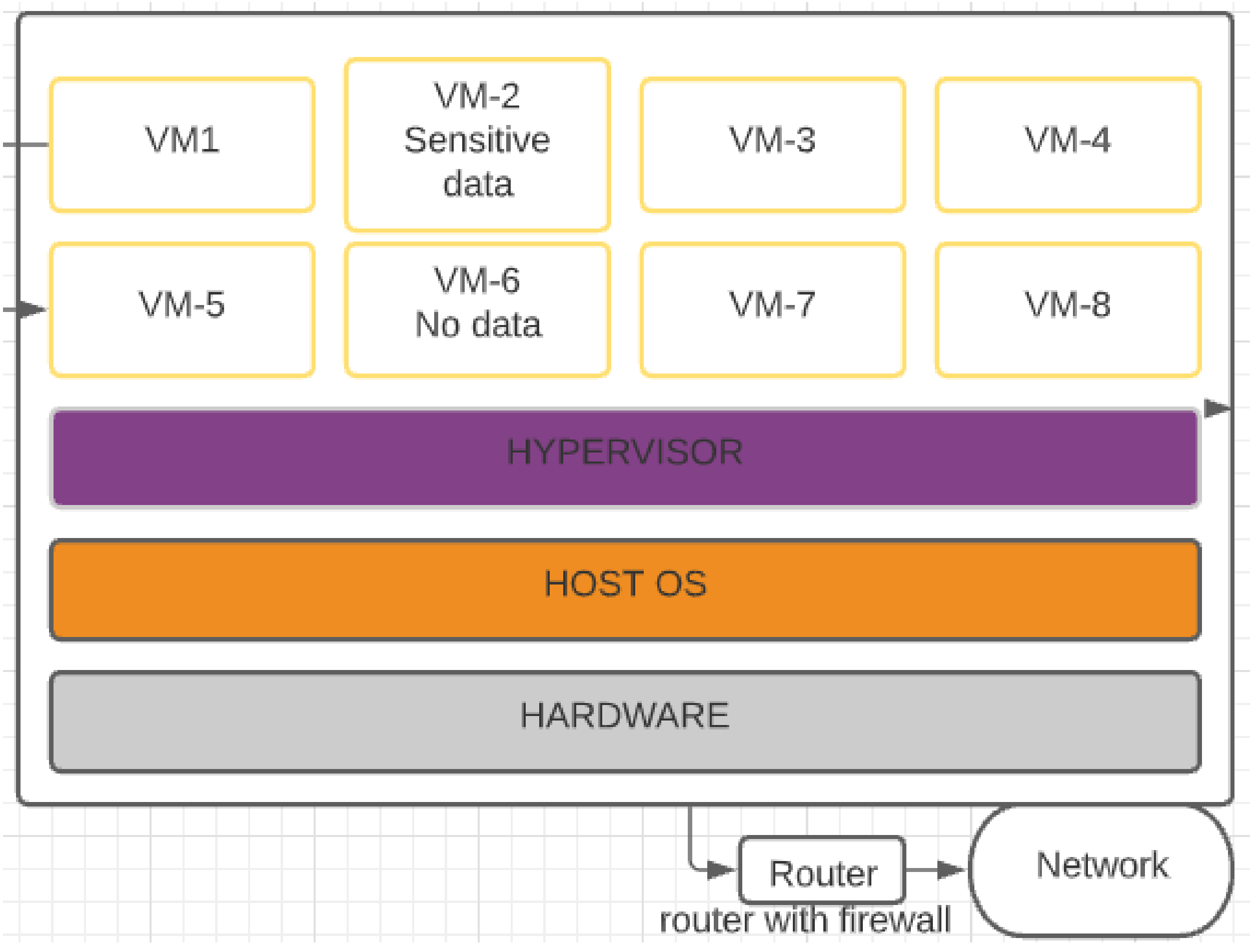


Diagram 3.2

Let us use diagram 3.2 to understand this security consideration. Imagine we have a network device such as a router configured with a firewall (security policy), and connected to our physical machine to help with connection to a network. It is important to realize if our firewall lacks the strength to identify virtual instances running on our physical machine, then it would not able to know when instances are communicating between themselves meaning the security measure it is supposed to deliver would not be achieved. This also means the firewall implemented must be configured and implemented with virtualization software to pick virtual instances in the physical machine.

Lack of Application Awareness

This is somehow similar to the contents in the blind spot security consideration: a lack of awareness scenario would be, if a network router is installed with firewall but does not possess a VM aware security policy or configuration attached to it, then it would result to the device not knowing what virtual instances are running in the physical machine and the instance that is responsible for the application trafficking from one point to another and vice versa. Again, for applications that are used for file sharing or peer to peer in our VM instances, security policies applied to our network devices might not be able to detect such activities and this can have a negative impact in an organization.

[17]

DOCKER

Docker can be seen as an advanced layer of virtualization that is known as a replacement of virtual machines. The beginning of this document included how virtualization works which involves utilizing a software called hypervisor which is a tool that incorporates the resources of a physical device (host) it operates within, to manage virtual machines (guests), installed in the hardware. Docker on the other hand is contrary. It is a tool that virtualizes or explores the resources of the operating system to support applications intended to be used.
It is a virtualization technology and application development tool that constitutes the creation and deployment of applications that are packaged in virtual containerized environments. Docker works with what is called containers. Containers can be understood to be microcomputers that are created with a docker tool known as docker engine and similar to hypervisors creating vms, this is how docker and containers can be depicted. Docker uses the operating system to share resources for these containers such as CPU, OS, memory, and network. Some huge benefits of using docker is it is lightweight and super fast.
[18]

BENEFITS OF DOCKER CONTAINERS

- Consistent and isolated environment: Docker creates environments that can promote the isolation of an application from another regardless of where it is deployed. With this benefit, applications remain consistent which contributes to high productivity such as less debug time, sufficient time for launching or testing features, and serviceability for user.
- Cost effectiveness with faster Deployment: Docker containers are popular with decreasing deployment sessions to seconds making it a valuable tool because from a traditional perspective, for example hypervisor-can consume so much time from day to days for provisioning, setting hardware up and running it and not to forget the extra work that can be involved in it. Docker makes this convenient because containers can be integrated with new apps making deployment swift and suitable.
- Mobility and ability to run anywhere: Docker images do not come with limitations which helps with deployment to be constant, movable, as well as scalable and with these, containers present the benefit of being able to run from anywhere as long as they are attached to an operating system in a clould environment. This is a huge advantage devop engineers appreciate.
- Repeatability and Automation: Docker makes it possible to reuse code that has been created due to the repeatable infrastructure and config it provides. This helps to speed development processes incredibly. It makes maintenance painless. Also, because of containerization, an application is isolated from other apps that run in the same system and therefore they do not conflict with themselves making it easy for maintenance.
- Test, Roll Back and Deploy: Docker images are easily versioned and due to the consistency, it creates in environments, it becomes easy to roll back if required. This is helpful whenever a problem occurs at any iteration of an image. This also helps in fulfilling CICD (continuous Integration and Continuous Deployment). Docker containers can hold internal configurations and dependencies and this helps in identifying setbacks.

- Flexibility: Docker enables engineers to create, test and release images across several servers and an intriguing thing about this tool is if an upgrade needs to be made within the release cycle of a product, all changes can be carried out to Docker containers followed with their test and finally their roll out. Also, Docker makes it possible to start and pause services, or modify apps fatly. Something useful when working in a cloud environment
- Collaboration, Modularity and Scaling: With Docker we can chunk an application to clean up, fix and restart without having to take down the entire application. It also enables engineers build an environment for applications that have small processes to communicate with themselves via APIs which devop engineers use for collaborative work in handling issues within good time.

[19]

Docker vs VMs

VM

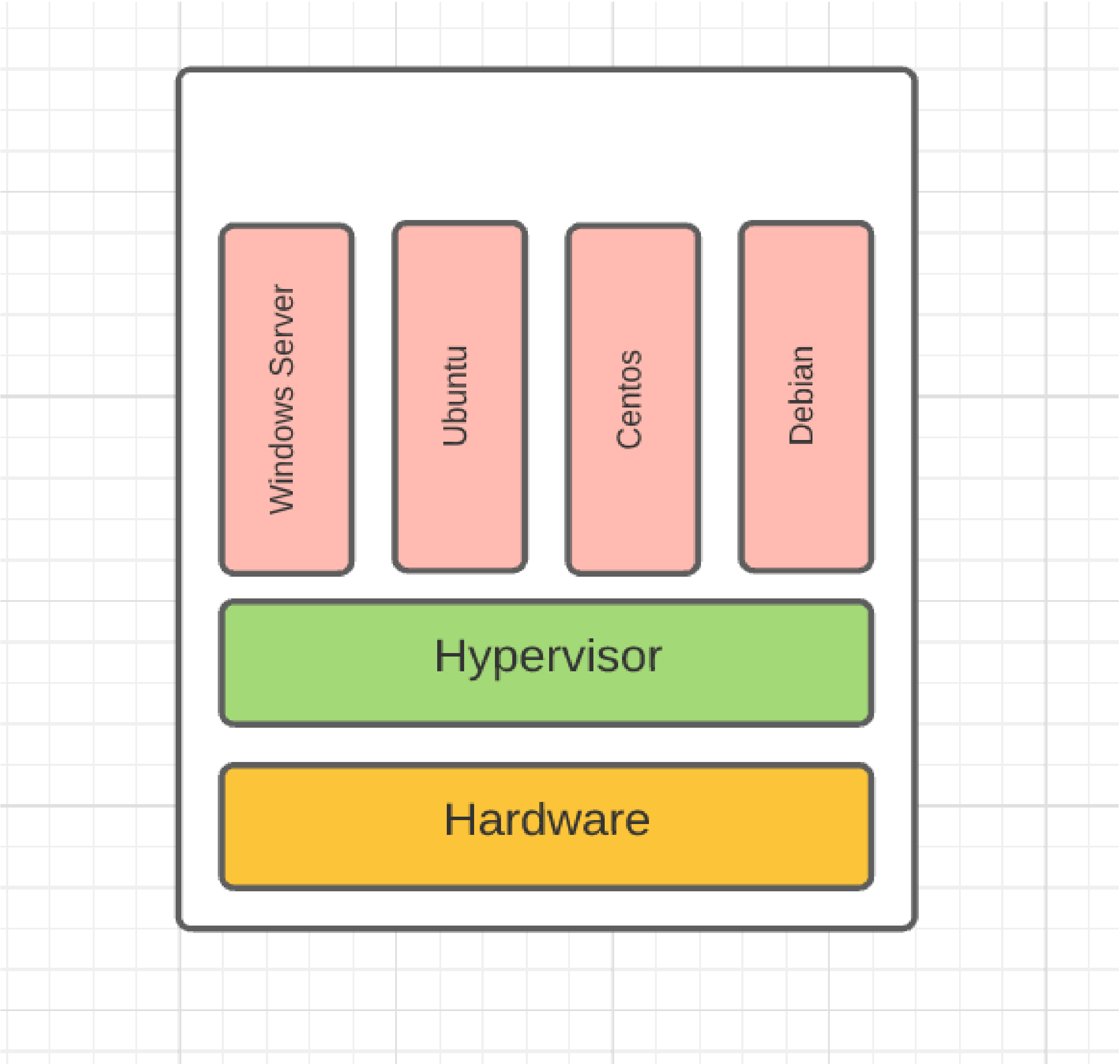


Diagram 4.0

Docker

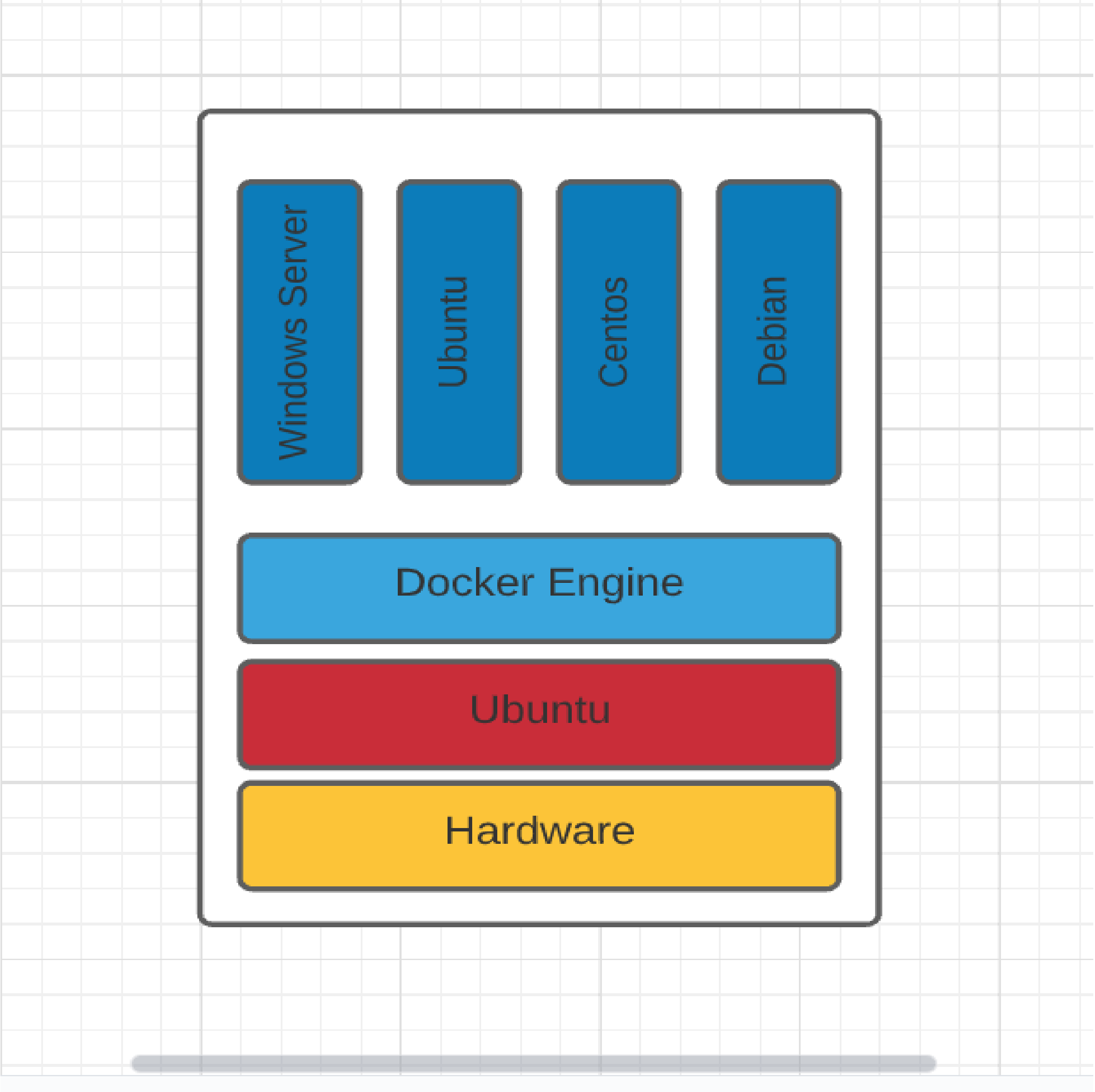


Diagram 4.1

VM	DOCKERS
Memory: VMs consume a lot of hardware memory resources and the downside to this is if we allocate an amount of our memory to several VMs on our hardware, we cannot use the memory space	Memory: Docker uses few computing resources in this area.
Portability: VMs would be on a disadvantage in this aspect because they each have their own OS and since OS cannot be ported to a different platform or function this will not be an advantage	Portability: Docker containers can be ported since they have inseparable OS's and are self-contained packages that execute required apps.
Boot-Time: Can take several minutes for VMs to boot	Boot-Time: Docker containers boot in seconds
Isolation: Interference is less due to efficient isolation mechanism	Isolation: Prone to adversities as no provision for isolation system
Deployment: Deployment is lengthy since instances are responsible for execution	Deployment: Single images are easy to deploy and containers can be utilized on different platforms
Usage: Tools are easy to use	Usage: Docker can be a bit complex because of its usage mechanisms comprising of third party and docker managed tools

[20]

Securing Docker containers

- Do not execute container as a root user: Containers are configured to be executed in root by default. This is so when building steps on the base image engineers need to install packages and create config settings and importantly, change user to be non-root user as soon as the configuration has been completed.
- Regularly update Docker and host: Ensure both Docker and host are up to date by initializing newest version. Also, ensure that updated operating system and container software prevent security issues that can occur. Take note that every update requires upgrades that facilitate the safety of the host and Docker.
- Namespaces: Processes running within a container cannot view, far less affect, processes running in another container or on the host system, thanks to namespaces.
- Each container will have its own virtual network, which implies that no container possesses privileged access to a different container's ports or interfaces. This does not mean interaction such as pinging between containers cannot exist, as long as the

host system is well configured accordingly, then communication between containers through their secure interfaces can occur.

- Configure resource quotas: Resource quotas configure each container by Docker and this helps to control various resources such as the memory and CPU which a container consumes. Implementing this measure would strengthen the efficiency of a Docker environment and can help to avoid any imbalance of resources that are allocated. This policy improves security around containers, and this can help to deny malicious code from spreading from one container into others because of the potentials quotas have to cut it off.
- Set container resource limits: Such a measure would help in reducing the potentials of containers to exhaust a great portion of a systems resources. By putting a limit to container use of resources, we can enhance security in an event of an attack.
- Control Groups: They use resource accounting and limitation techniques. They not only give many useful data, but help each container receive its fair amount of memory, CPU, and disk I/O; and, very significantly, a container would not affect the system by depleting one of the many resources. They're especially crucial for cross systems, such as public and private PaaS, to ensure optimal durability (and efficiency) regardless of if various apps act otherwise.
- Secure container registers: Container registry helps to store and provide images to build containers and with this, we can build a centralized repository to enable download of these images swiftly. To prevent security risk, it is good practice to install a mechanism to enhance security on registers. Docker Trusted Registry is a helpful tool to accomplish this especially since it is installed behind a firewall, thereby preventing the possibility of breaches in a network.
- Monitor API and network security: Communication between containers is important in Docker and API and network tools help to achieve such interaction. It is good practice to ensure these tools are integrated with Docker fully monitored, configured, and updated to maintain security measures against any compromise.

[21]
[22]

PRICING

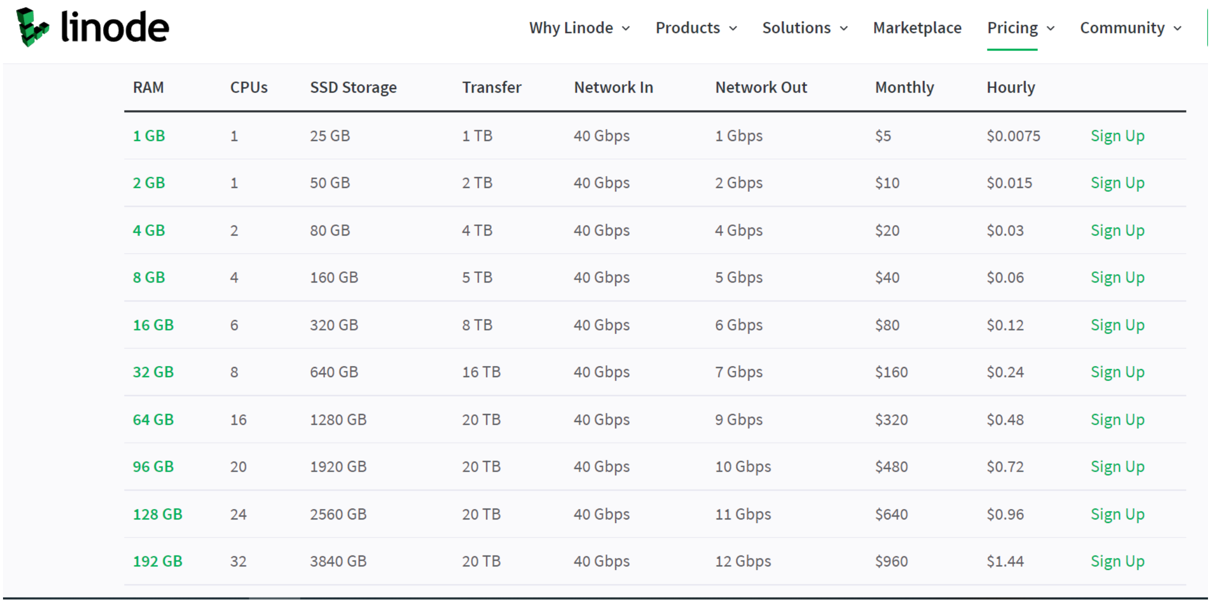
Instance Types

- Shared CPU Instances
- Dedicated CPU Instances
- High Memory Instances
- GPU Instances

Shared CPU Instances

1 GB - 192 GB Memory, 1 - 32 Shared vCPU Cores, 25 GB - 2840 GB Storage
Starting at $5/Mo ($0.0075/hour)
Includes

- Medium to low traffic websites, such as for marketing content and blogs
- Forums
- Development and staging servers
- Low traffic databases
- Worker nodes within a container orchestration cluster

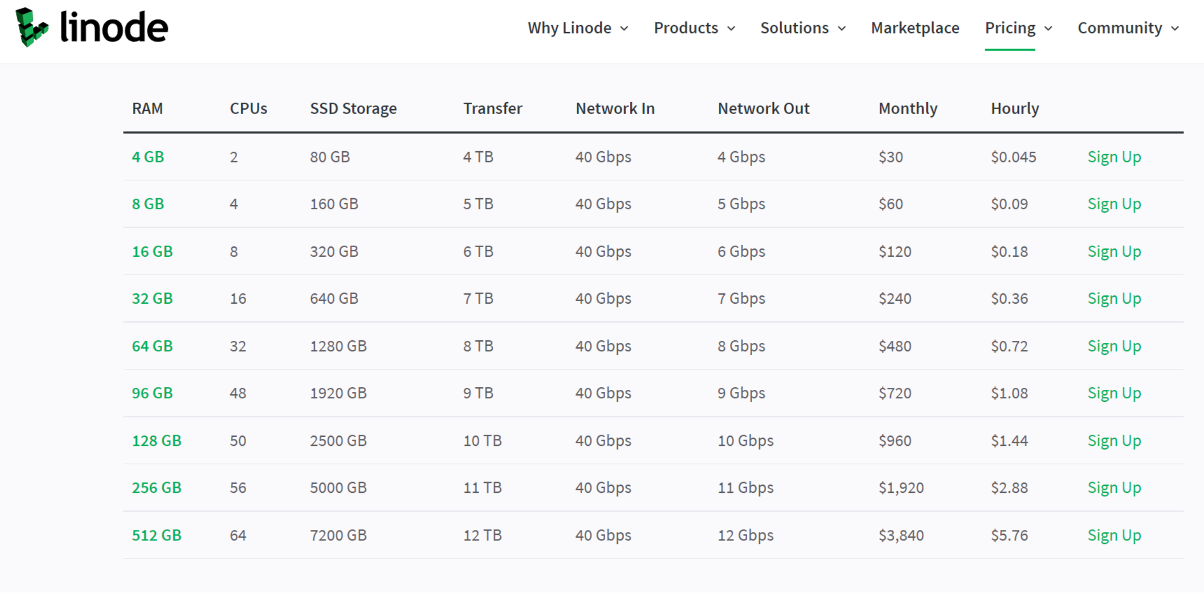


RAM	CPUs	SSD Storage	Transfer	Network In	Network Out	Monthly	Hourly	
1 GB	1	25 GB	1 TB	40 Gbps	1 Gbps	$5	$0.0075	Sign Up
2 GB	1	50 GB	2 TB	40 Gbps	2 Gbps	$10	$0.015	Sign Up
4 GB	2	80 GB	4 TB	40 Gbps	4 Gbps	$20	$0.03	Sign Up
8 GB	4	160 GB	5 TB	40 Gbps	5 Gbps	$40	$0.06	Sign Up
16 GB	6	320 GB	8 TB	40 Gbps	6 Gbps	$80	$0.12	Sign Up
32 GB	8	640 GB	16 TB	40 Gbps	7 Gbps	$160	$0.24	Sign Up
64 GB	16	1280 GB	20 TB	40 Gbps	9 Gbps	$320	$0.48	Sign Up
96 GB	20	1920 GB	20 TB	40 Gbps	10 Gbps	$480	$0.72	Sign Up
128 GB	24	2560 GB	20 TB	40 Gbps	11 Gbps	$640	$0.96	Sign Up
192 GB	32	3840 GB	20 TB	40 Gbps	12 Gbps	$960	$1.44	Sign Up

Dedicated CPU Instances

4 GB - 512 GB Memory, 2 - 64 Dedicated vCPUs, 80 GB - 7200 GB Storage
Starting at $30/Mo ($0.045/hour).
Includes

- CI/CD toolchains and build servers
- Game servers (like Minecraft or Team Fortress)
- Audio and video transcoding
- Big data (and data analysis)
- Scientific computing
- Machine learning and AI
- High Traffic Databases (Galera, PostgreSQL with Replication
- Manager, MongoDB using Replication Sets)
- Replicated or Distributed Filesystems (GlusterFS, DRBD)

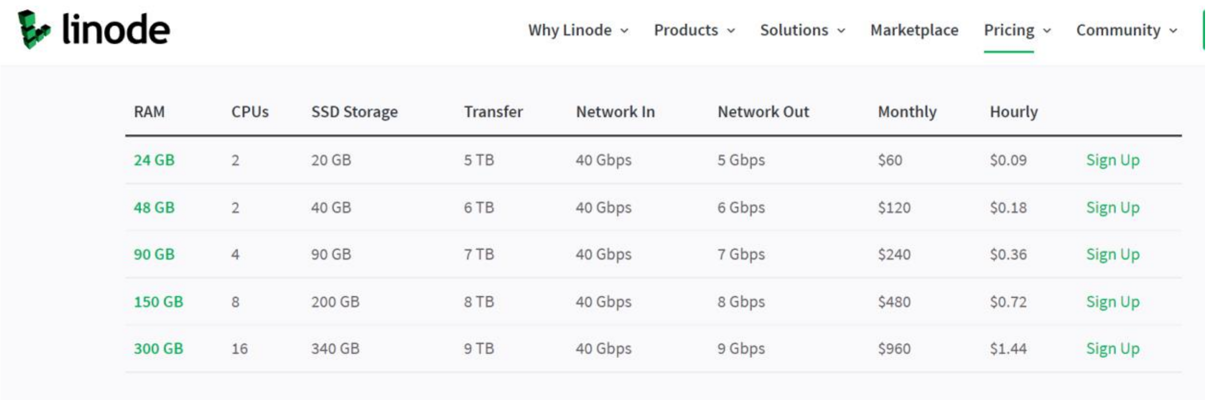


RAM	CPUs	SSD Storage	Transfer	Network In	Network Out	Monthly	Hourly	
4 GB	2	80 GB	4 TB	40 Gbps	4 Gbps	$30	$0.045	Sign Up
8 GB	4	160 GB	5 TB	40 Gbps	5 Gbps	$60	$0.09	Sign Up
16 GB	8	320 GB	6 TB	40 Gbps	6 Gbps	$120	$0.18	Sign Up
32 GB	16	640 GB	7 TB	40 Gbps	7 Gbps	$240	$0.36	Sign Up
64 GB	32	1280 GB	8 TB	40 Gbps	8 Gbps	$480	$0.72	Sign Up
96 GB	48	1920 GB	9 TB	40 Gbps	9 Gbps	$720	$1.08	Sign Up
128 GB	50	2500 GB	10 TB	40 Gbps	10 Gbps	$960	$1.44	Sign Up
256 GB	56	5000 GB	11 TB	40 Gbps	11 Gbps	$1,920	$2.88	Sign Up
512 GB	64	7200 GB	12 TB	40 Gbps	12 Gbps	$3,840	$5.76	Sign Up

High Memory Instances

24 GB - 300 GB Memory, 2 - 16 Dedicated vCPUs, 20 GB - 340 GB Storage
Starting at $60/Mo ($0.09/hour).
Includes

- Any production application that requires large amounts of memory
- Redis and Memcached are in-memory database caching techniques. These programs provide extremely quick data retrieval, however they do so in a non-persistent manner (with some caveats). As a result, they're frequently utilized in conjunction with a separate instance of a persistent database server.
- In-memory databases, like those made feasible by NoSQL and other approaches
- Processing of large amounts of data (and data analysis)

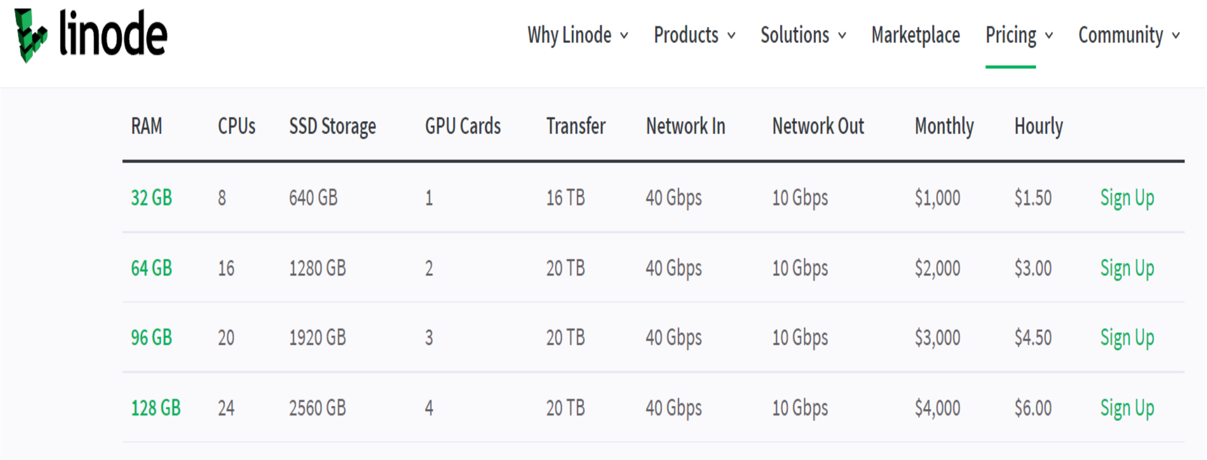


RAM	CPUs	SSD Storage	Transfer	Network In	Network Out	Monthly	Hourly	
24 GB	2	20 GB	5 TB	40 Gbps	5 Gbps	$60	$0.09	Sign Up
48 GB	2	40 GB	6 TB	40 Gbps	6 Gbps	$120	$0.18	Sign Up
90 GB	4	90 GB	7 TB	40 Gbps	7 Gbps	$240	$0.36	Sign Up
150 GB	8	200 GB	8 TB	40 Gbps	8 Gbps	$480	$0.72	Sign Up
300 GB	16	340 GB	9 TB	40 Gbps	9 Gbps	$960	$1.44	Sign Up

GPU Instances

Starting at $1000/Mo ($1.50/hour).
Includes

- Machine Learning and AI
- Big Data
- Video Encoding
- General Purpose Computing Using NVIDIA’s CUDA Toolkit
- Graphics Processing



RAM	CPUs	SSD Storage	GPU Cards	Transfer	Network In	Network Out	Monthly	Hourly	
32 GB	8	640 GB	1	16 TB	40 Gbps	10 Gbps	$1,000	$1.50	Sign Up
64 GB	16	1280 GB	2	20 TB	40 Gbps	10 Gbps	$2,000	$3.00	Sign Up
96 GB	20	1920 GB	3	20 TB	40 Gbps	10 Gbps	$3,000	$4.50	Sign Up
128 GB	24	2560 GB	4	20 TB	40 Gbps	10 Gbps	$4,000	$6.00	Sign Up

[23]

ANSIBLE

Automation help virtualized environments undergo quick deliveries with regards to deployment, installation, configuration, and maintenance. A key word to describe automation is simplicity and because of this, IT engineers continue to achieve a lot of goals. One tool that has been helping these experts is Ansible.

Ansible is an IT automation tool that helps with various development processes such as configure systems, deploy software, and orchestrate several IT tasks.

How it works

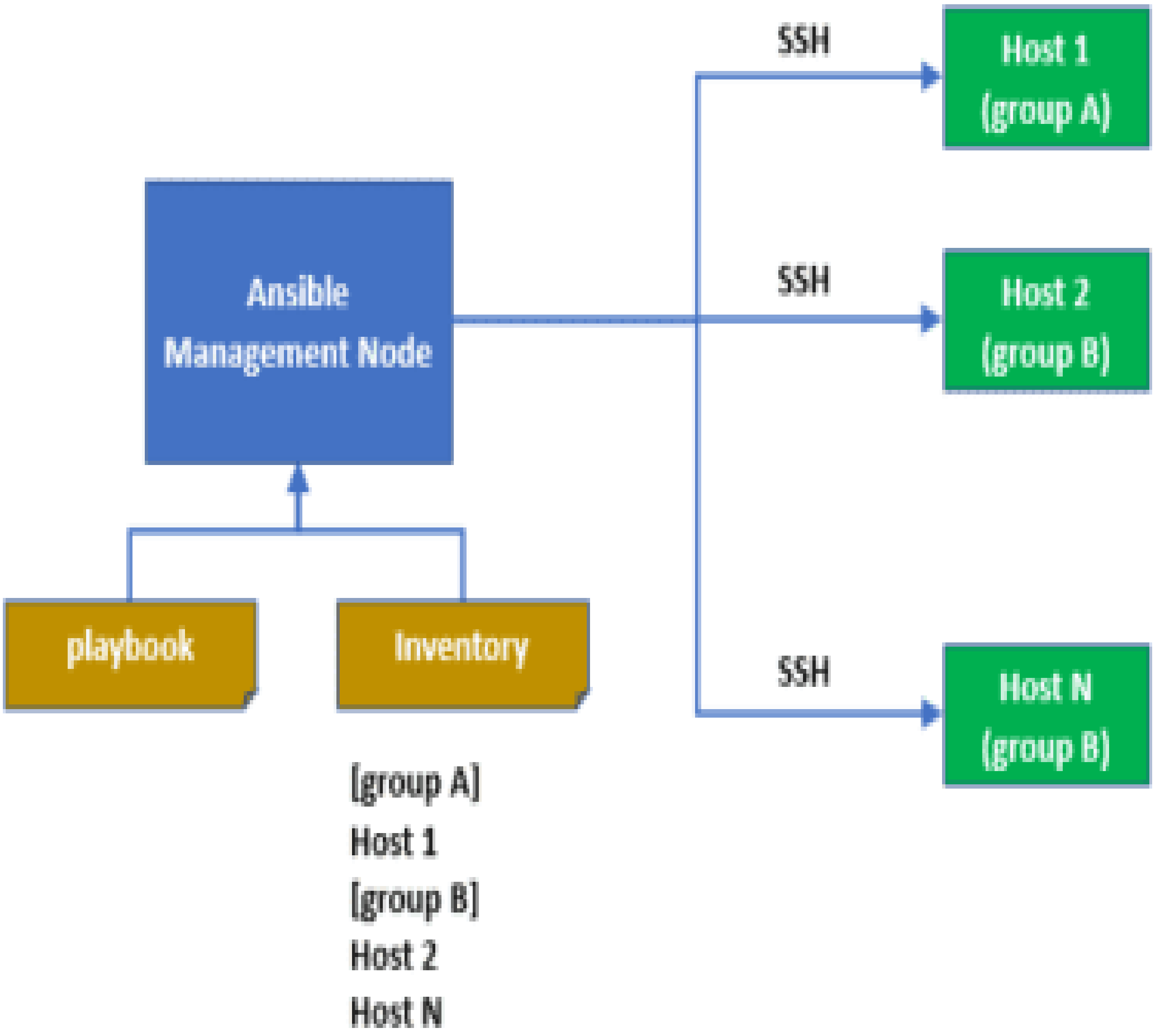


Diagram 6.0

COMPONENT	DESCRIPTION
Management or Control Node	This is where ansible is installed and used to manage nodes (servers)
Inventory	This is where INI file that ansible manages will be stored
Playbook	Holds the YAML file with instructions to be automated
Task	Defines all procedures that would be executed
Module	Build ansible tasks
Role	A collection of playbooks, templates and various files that can be reused or shared
Play	Executes playbooks from start to end
Facts	Information about ansible targets the are collected by nodes
Handlers	Used to restart or reload a service

[24]

[25]

CONCLUSION

Based on findings from this research, the impact of virtualization is incredibly huge in IT operations. Not only has it redefined how we accomplish tasks but the systematic nature that it creates, balancing two elements, consisting of hardware and software is without a doubt an appreciative phenomenon. Docker containers are very secure and reliable by design, especially when running an instance as a non-privileged user inside a container. This technology opens a discussion about the advancements it continues to make in form of comparisons between its types. There are certain questions to ask when implementing a virtual environment.

What is it going to be used for? Is it for an SME and if so, should there be need to anticipate growth? Just a few questions among several but the goal is to ensure it satisfies the demands of the people. A hypervisor can be utilized but consuming hardware resources would be a pillar. Containerization presents very productive ways, but security remains an issue. Though the benefits are remarkable, this technology based on discoveries remains key in IT operations.

REFERENCES

[1]
Tech Differences. 2021. *Difference Between Full Virtualization and Paravirtualization (with Comparison Chart) - Tech Differences*. [online] Available at: <https://techdifferences.com/difference-between-full-virtualization-and-paravirtualization.html> [Accessed 26 November 2021].

[2]
Networksandservers.blogspot.com. 2021. *<h1>Operating System-Level Virtualization Explained</h1>*. [online] Available at: <https://networksandservers.blogspot.com/2011/11/this-kind-of-server-virtualization-is.html> [Accessed 26 November 2021].

[3]
2021. [online] Available at: <https://www.researchgate.net/figure/Para-virtualization-concepts_fig2_268291860> [Accessed 26 November 2021].

[4]
Tech Differences. 2021. *Difference Between Full Virtualization and Paravirtualization (with Comparison Chart) - Tech Differences*. [online] Available at: <https://techdifferences.com/difference-between-full-virtualization-and-paravirtualization.html> [Accessed 26 November 2021].

[5]
Networksandservers.blogspot.com. 2021. *<h1>Operating System-Level Virtualization Explained</h1>*. [online] Available at: <https://networksandservers.blogspot.com/2011/11/this-kind-of-server-virtualization-is.html> [Accessed 26 November 2021].

[6]
Docs.microsoft.com. 2021. *Welcome to Remote Desktop Services in Windows Server 2016*. [online] Available at: <https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/welcome-to-rds> [Accessed 26 November 2021].

[7]
Wikibon.org. 2021. *Network Function Virtualization Or NFV Explained - Wikibon*. [online] Available at: <http://wikibon.org/wiki/v/Network_Function_Virtualization_or_NFV_Explained> [Accessed 26 November 2021].

[8]
2021. [online] Available at: <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor> [Accessed 26 November 2021].

[9]
HowStuffWorks, Tech, Computer, Hardware and Networking, 2021. *How Server Virtualization Works*. [online] HowStuffWorks. Available at: <https://computer.howstuffworks.com/server-virtualization.htm> [Accessed 26 November 2021].

[10]
2021. [online] Available at: <https://www.redhat.com/en/topics/virtualization/what-is-virtualization> [Accessed 26 November 2021].

[11]
CircleCI. 2021. *Top 6 benefits of virtualization*. [online] Available at: <https://circleci.com/blog/top-6-benefits-of-virtualization/#c-consent-modal> [Accessed 26 November 2021].

[12]
Youtube.com. 2021. [online] Available at: <https://www.youtube.com/watch?v=MMiubSial6E> [Accessed 26 November 2021].

[13]
Azure.microsoft.com. 2021. *What is a Public Cloud - Definition | Microsoft Azure*. [online] Available at: <https://azure.microsoft.com/en-us/overview/what-is-a-public-cloud/> [Accessed 26 November 2021].

[14]
Azure.microsoft.com. 2021. *What is a Private Cloud - Definition | Microsoft Azure*. [online] Available at: <https://azure.microsoft.com/en-us/overview/what-is-a-private-cloud/> [Accessed 26 November 2021].

[15]
Azure.microsoft.com. 2021. *What is Hybrid Cloud Computing – Definition | Microsoft Azure*. [online] Available at: <https://azure.microsoft.com/en-us/overview/what-is-hybrid-cloud-computing/> [Accessed 26 November 2021].

[16]
Youtube.com. 2021. [online] Available at: <https://www.youtube.com/watch?v=AjtdZ3gFRjU> [Accessed 26 November 2021].

[17]
Youtube.com. 2021. [online] Available at: <https://www.youtube.com/watch?v=hjKs0AJTIX0&t=180s> [Accessed 26 November 2021].

[18]
Youtube.com. 2021. [online] Available at: <https://www.youtube.com/watch?v=eGz9DS-aIeY&t=425s> [Accessed 26 November 2021].

[19]
Uros Pavlovic, G., 2021. *Docker Containers: Top 7 Benefits of Docker Containerization*. [online] Hentsu. Available at: <https://hentsu.com/docker-containers-top-7-benefits/> [Accessed 26 November 2021].

[20]
Arora, 2021. *Docker vs. Virtual Machines: Differences You Should Know*. [online] Cloud Academy. Available at: <https://cloudacademy.com/blog/docker-vs-virtual-machines-differences-you-should-know/> [Accessed 26 November 2021].

[21]
Engineering Education (EngEd) Program | Section. 2021. *Docker Security - Best Practices to Secure a Docker Container*. [online] Available at: <https://www.section.io/engineering-education/best-practices-to-secure-a-docker-container/> [Accessed 26 November 2021].

Docker Documentation. 2022. *Docker security*. [online] Available at: <https://docs.docker.com/engine/security/> [Accessed 14 April 2022].

[22]
Youtube.com. 2021. [online] Available at: <https://www.youtube.com/watch?v=JE2PJbbpjsM> [Accessed 26 November 2021].

[23]
2022. [online] Available at: <https://www.linode.com/docs/guides/choosing-a-compute-instance-plan/#shared-cpu-instances> [Accessed 15 April 2022].

[24]
Google.com. 2021. *ansible explanation - Google Search*. [online] Available at: <https://www.google.com/search?q=ansible+explanation&sxsrf=AOaemvJRH5zxyR-Lmm0DodX_6uWATVoIeA:1637948919442&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjOi4Ocy7b0AhVGQMAKHYSjBhMQ_AUoAXoECAEQAw&biw=1920&bih=969&dpr=1#imgrc=Ty7b5PFFC9VvWM&imgdii=RNVtGGhPe1E70M> [Accessed 26 November 2021].

[25]
Theseus.fi. 2021. [online] Available at: <https://www.theseus.fi/bitstream/handle/10024/343383/Ronald_Clark_Thesis.pdf?sequence=2> [Accessed 26 November 2021].