Review Article

iMedPub Journals
http://www.imedpub.com

American Journal of Computer Science and Engineering Survey

2020
Vol. 8 No.2:10

DOI: 10.36648/computer-science-engineering-survey.08.02.10

Implementation of Inter-Networking with Host Internet in Oracle® VirtualBox Guest Virtual Machines

Robert Method Karamagi[1*] and Dr. Said Ally[2]

[1]Department of Information & Communication Technology, The Open University of Tanzania, Dar es salaam, Tanzania

[2]Department of Science, Technology and Environmental Studies, The Open University of Tanzania, Dar es salaam, Tanzania

***Corresponding author:** Robert Method Karamagi, Department of Information & Communication Technology, The Open University of Tanzania, Dar es salaam, Tanzania, E-mail: robertokaramagi@gmail.com

Citation: Karamagi RM, Ally S (2020) Implementation of Inter-Networking with Host Internet in Oracle® VirtualBox Guest Virtual Machines. Am J Comput Sci Eng Surv Vol. 8 No. 2: 10.

Abstract

Oracle® VirtualBox is an open-source hosted hypervisor for virtualization, developed by the Oracle Corporation. For this paper, we have created three virtual machines (VMs), each with different operating systems, assigned specific IP addresses for each VM and the host operating system. We have looked into the reasons for the choice of the IP class and networking mode among the Bridged, NAT or Host-only choice, then compared the networking modes based on the access between the VM and Host, VM and VM, and VM and LAN or external link.

Keywords: Oracle® VirtualBox; Network Address Translation (NAT); Host-only adapter; Dynamic Host Configuration Protocol (DHCP)

Received: October 02, 2020; **Accepted:** October 16, 2020; **Published:** October 23, 2020

Introduction

Oracle® VM VirtualBox is virtualization software that is cross-platform. It allows users to run multiple operating systems simultaneously by an extension of their current computer. It is designed for developers and IT professionals. Oracle® VM VirtualBox is compatible with Microsoft® Windows, Linux, Mac OS X, and Oracle® Solaris. It is an ideal environment for launching tests, development, demonstration and deployment of solutions across many platforms on a single machine.

It is lightweight, easy to install, use, and was crafted to apply innovations brought up in the x86 hardware platform. Despite the simple exterior, it is a very fast and powerful engine for virtualization with a well-earned reputation for having great speed and agility. Oracle® VM VirtualBox has a number of features that are innovative to produce efficient business gain. Among them include, high performance, powerful virtualization, and a vast range of guest operating systems that are supported [1].

As a bridge to open source and cloud development, users can create and deploy their virtual machines almost anywhere. They have the capability to upload and download to and from the cloud. They can also perform their reviews and commit changes while offline.

Oracle® VM VirtualBox is the most popular open source and free cross-platform virtualization software in the world with over thousands of daily downloads.

It has world-class support and development supplied by Oracle® in conjunction with a vibrant community participation [2].

Main features of VirtualBox

The main features of VirtualBox are below:

Portability: A large number of 32-bit and 64-bit host operating systems can be run on VirtualBox. Virtual machines that are created on different hosts and/or with virtualization software that is different can be run.

Multiple virtualization interfaces: Three different interfaces of virtualization are provided by VirtualBox.

- Minimal-The presence of an environment that is virtualized is announced.
- KVM- A Linux KVM hypervisor interface that is recognied by Linux kernels is presented. This interface is chosen for implementations and tests/
- Hyper-V- A Microsoft® Hyper-V hypervisor interface is presented that can be recognized by Windows 7 and later operating systems.

Multiple frontends: VirtualBox provides a user interface called a

frontend, such as:

• VBox Manage- Advanced settings for VMs are allowed by this textual interface.

• VirtualBox-This is the default frontend, which is based on Qt.

• VBoxSDL-It is based on SDL and provides an alternative option for the frontend. It is useful for testing during development and for business use. Vbox Manage is then used to control the VMs.

• VBoxFB-This is a GUI that sits directly on the Linux frame buffer known as the "Frame buffer GUI". It is currently not maintained.

Hardware virtualization is not required: Processor features such as Intel VT-x or AMD-V are not required by VirtualBox, even though there is full support of the virtualization of hardware. This makes VirtualBox capable of being used on old hardware which does not have any of these features.

Guest additions: VirtualBox Guest Additions is a package of software that provides additional communication and integration, such as, automated video resolution adjustment, 3D accelerated graphics, and more, with the host system. It can be installed inside of the guest systems that are supported.

Great hardware support: VirtualBox supports the following and more:

• Guest multiprocessing (Symmetric multiprocessing (SMP)). Regardless of the number of CPU cores that are physically present on the host, up to 32 virtual CPUs can be presented by VirtualBox to each virtual machine.

• USB device support. Without the need of installing device-specific drivers on the host, arbitrary USB devices are allowed to connect to the VMs, by the virtual USB controller implemented by VirtualBox.

• Hardware compatibility. A vast array of virtual devices is capable of being virtualized by VirtualBox. Among these, include, a variety of virtual network cards (e1000 being present), SATA hard disk controllers, IDE, SCSI, etc.

• Full ACPI support. VirtualBox fully supports the Advanced Configuration and Power Interface (ACPI).

Multi generation branched snapshots: Arbitrary snapshots of the VM state can be saved by VirtualBox. It is possible to revert the VM to any snapshot, by going back in time, and an alternative VM configuration can hence by started from there, which effectively creates a whole snapshot tree.

VM groups: A groups feature is provided by VirtualBox. Users are enabled to organize and control VMs individually and collectively.

Clean and modular architecture: VirtualBox provides a clean distinction between the client and server code (i.e. the code related to the Virtual Machine Manager (VMM) and that related to the VMs respectively). Its design is extremely modular and has internal programming interfaces that are well-defined.

VirtualBox components

As the Graphical User Interface (GUI) of VirtualBox is opened with at least one initiated VM, three processes are running.

VBoxSVC: This process automatically starts when the first client process is initiated. It is a background process that is always running. After the last client exits the process also terminates after a short time. The duty of this service is to maintain the state of all the VMs. It is also known as the server process.

The GUI process: This process is used to communicate the state changes and settings to the VBoxSVC. Another name for it is the client process.

The hypervisor process: All the registered VMs are initialized, and the VBoxSVC component's execution starts, when VirtualBox is launched (e.g. the GUI of VirtualBox). Before the start of the execution, when a VM is launched, the server is requested for all the settings details by the client. This information is delivered to the VMM process that is initiating. The hypervisor has the ability to execute the VM, finally.

VirtualBox kernel modules

Different kernel modules are provided by VirtualBox that the user should add to the host kernel:

• vboxdrv-The VMM uses this to gain control over the host system. It is the only mandatory module. The host/guest world switches and device emulations are managed by it.

• vboxnetadp-This is the short form of "VirtualBox Network Adapter". It is used in the creation of vboxnet, a host networking interface. The interface is fundamentally a virtual switch which connects VMs to the host and/or to each other. When the networking mode in VirtualBox is set to bridged or host-only, this kernel module is needed.

• vboxnetflt-This module stands for "VirtualBox Network Filter". It injects and filters packets and attaches itself to the host on a real interface. Similar tovboxnetadp, its necessity lies for only the bridged and host-only modes.

• vboxpci-PCI card pass through is provided by this module. When the driver that is related is not present on the host, when the user wants to use a PCI device on the guest, this module may come to use.

Literature Survey

Oracle® VirtualBox gives up to eight virtual PCI Ethernet cards for each virtual machine. For each such card, you can separately choose the accompanying:

• The hardware that is intended to be virtualized.

• The virtualization mode in which the virtual card shall operate in, with regard to your physical networking hardware on the host.

In the Network section of the Settings dialog, in the graphical user interface of Oracle® VirtualBox, the network cards can be configured. You may likewise configure the network cards via command line using VBox Manage modifyvm [3].

Figure 1 shows various network adapters on Oracle® VirtualBox.

For each card, you can individually select what kind of hardware will be presented to the virtual machine.

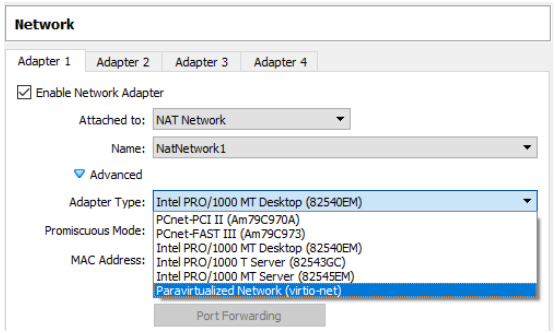

Figure 1: Various network adapters on Oracle® virtualbox.

Oracle VM VirtualBox can virtualize the following types of networking hardware:

• AMD PCNet PCI II (Am79C970A)

• AMD PCNet FAST III (Am79C973)

• Intel PRO/1000 MT Desktop (82540 EM)

• Intel PRO/1000 T Server (82543 GC)

• Intel PRO/1000 MT Server (82545 EM)

• Paravirtualized network adapter (virtio-net)

Networking modes

• Not attached-The presence of a network card is reported to be present to the Guest by Oracle® VirtualBox, but a connection is not available. It is similar to no Ethernet cable being plugged in. When you use this mode, the virtual Ethernet cable may be pulled and the connection disrupted. This has the application of letting the Guest Operating System know that a network connection is unavailable and it should enforce a reconfiguration.

• Network Address Translation (NAT)-This mode is suitable when you want to have a working Internet connection inside the guest.

• NAT network-This is an internal type of network that allows outbound connections.

• Bridged networking-When advanced networking needs, such as network simulations and running servers in a guest is required, this network can be enabled. A connection to one of your network cards will be initiated and network packets will be exchanged directly. This shall circumvent the network stack of your host operating system.

• Internal networking-A different type of software-based network that is visible to selected virtual machines can be created. It will not be visible to applications running on the host or those that run in the outside world.

• Host-only networking-A network that contains the host along with a set of virtual machines can be created. There is no need of the physical network interface of the host. A virtual network interface, similar to a loopback interface, is made on the host that gives a connection throughout the virtual machines and the host.

• Generic networking-This is a rarely used mode, whereby the same generic network interface is shared. A user is allowed to select a driver. This driver can be included either with Oracle® VirtualBox or be in an extension pack distribution.

Sub-modes are available, listed below:-

UDP Tunnel-Virtual machines that run on separate hosts may be interconnected directly with ease and transparency, over a network infrastructure that is existing.

VDE (Virtual Distributed Ethernet) networking-A Virtual Distributed Ethernet switch may be connected on either a Linux or FreeBSD host.

Table 1 shows a summary of the main networking modes.

Mode	VM→ Host	VM→ Host	VM1↔ VM2	VM→ Net/LAN	VM← Net/LAN
Host-only	+	+	+	-	-
Internal	-	-	+	-	-
Bridged	+	+	+	+	+
NAT	+	Port forward	-	+	Port forward
NAT service	+	Port forward	+	+	Port forward

Table 1: Overview of Networking Modes.

Network address translation service

The Network Address Translation (NAT) service and a home router work in quite a similar way. The systems that use it are grouped into a network which prevents the direct access of systems inside this network from those outside of it. It allows the systems inside of it to communicate with themselves and with other systems that are outside it, using TCP and UDP over IPv4 and IPv6.

The NAT service is attached to an internal network. The virtual machines that use it are attached to the internal network. When the NAT service is created, the name of the internal network is chosen. The creation of the internal network will take place if it does not exist already [4].

Host-only networking

The Host-Only networking mode may be depicted as a hybrid version of the bridges and internal networking modes. The bridged networking capability lies in the communication of virtual machines with their host as if a physical Ethernet switch was connected through them. The internal networking feature is the fact that a physical networking interface does not have to be there, and communication of the virtual machines and the world outside the host does not take place since a physical networking interface is not connected to them.

Oracle® VirtualBox creates a new software interface on the host, when the host-only networking is used. This interface appears next to the networking interfaces that already exist. In the case of bridged networking, a physical interface that exists is used for attaching virtual machines. Whereas, a new loopback interface is created on the host, in the host-only networking. Traffic that is not seen between the virtual machines, in the internal networking, can be intercepted on the loopback interface on the host.

Virtual appliances that are preconfigured, where multiple machines are shipped together and designed to cooperate, find host-only networking useful. An example is where a web server

may be on one virtual machine and a database on the other. Since the two are meant to communicate, then Oracle® VirtualBox can be instructed by the appliance to set up a host-only network for the two machines.

Bridged networking

A device driver that is available on your host system which filters data from your physical network adapter is used by Oracle® VirtualBox in the bridged networking mode. Hence, the driver is given the name, net filter driver. Oracle® VirtualBox is enabled to intercept data from the physical network and then inject data in it. This creates a new network interface effectively in software. The host system sees the guest as if it is connected to the interface physically using a network cable, when the guest uses such a new software interface. Data can be sent by the host to the guest through that interface and may similarly receive data from it. Routing or Bridging may thus be set up between the guest and the rest of the network [5].

Software virtualization

Hardware virtualization is fully supported by VirtualBox. It is not mandatory, but in the event, it is not available, Software Virtualization is used. To clearly grasp the concept behind software virtualization, an understanding of the mechanism in which Central Processing Units (CPUs) provide microcode level protection, called Protection Rings or Privilege Rings is required.

Four privilege levels or rings are present. They are numbered from 0 to 3. The most privileged ring is Ring 0 and the least privileged is Ring 3.

The execution of privileged instructions or access of data is restricted by the system software when using the rings. Some device drivers and the operating system (OS) run in Ring 0, in a majority of the environments and the applications run in Ring 3.

There is a need to distinguish between the host and guest context [6]. Protection rings is shown in **Figure 2**.

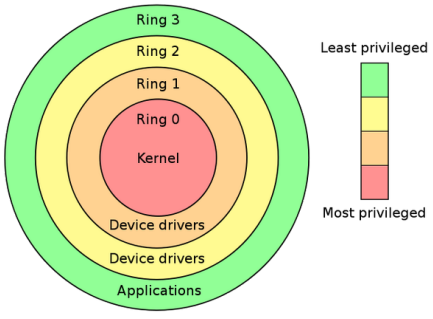

Figure 2: Protection rings.

• All things are like there is no VMM that is active in the host context. If a different application on the host has CPU time scheduled to it, this may be the mode that is active. There is both a host Ring 3 and Ring 0 mode, in that case [7].

• The VM is active in the guest context. With the guest code being run in Ring 3, the page tables may be set properly by the hypervisor and the processor may natively run that code. A challenge comes when intercepting what is done by the guest's kernel.

The host is set up by VirtualBox through its Ring 0 support kernel driver, when starting a VM. This allows it to natively run most of the guest code, and at the "bottom" of the picture, insert itself. Control can then be assumed when needed. E.g. during the execution of a privileged instruction, a trap is done by the guest; this may then be handled by VirtualBox and either a request may be routed to a virtual device or possibly the guest or host OS may be delegated handling.

Methodology

We have installed three guest virtual machines on VirtualBox:-

• Kali Linux 2020.3

• Windows 10 Education

• Windows Server 2019 Standard

Installed guest virtual machines are shown in **Figure 3**.

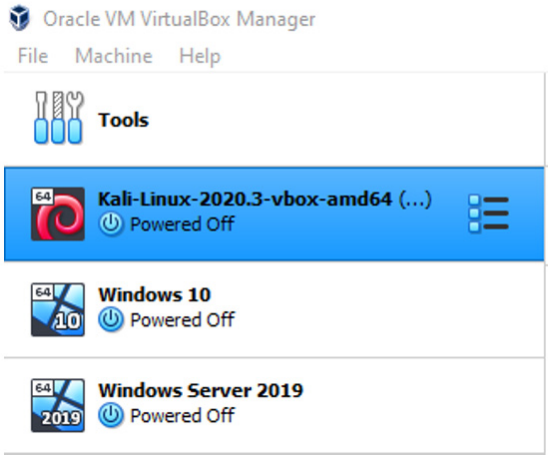

Figure 3: Installed guest virtual machines.

NAT network implementation

To get Internet connectivity from the host as well as VM to VM communication, we set up a NAT Network.

The following is a command to create a NAT network and attach a DHCP server using the command prompt:

C:\Program Files\Oracle\VirtualBox>VBoxManage natnetwork add-net name NatNetwork1--network "10.0.2.0/24"--enable--dhcp on

The name of the internal network is NatNetwork1, and the network address and mask of the NAT service interface is 10.0.2.0/24. The default setting for this static configuration is a gateway assignment of 10.0.2.1, although it is subject to change.

You may start the NAT service in the command prompt as follows:

C:\Program Files\Oracle\VirtualBox>VBoxManage natnetwork start--netname NatNetwork1

The list of registered NAT networks is seen with the command below:

To see the list of registered NAT networks, use the following command:

C:\ProgramFiles\Oracle\VirtualBox>VBoxManagelist natnetworks

In the Oracle® VirtualBox Preferences, under the Network section, you may find the Details of the NatNetwork1. You may similarly, create it from the interface.

Figure 4 shows Listing NAT networks and **Figure 5** shows Configuring a NAT network on Oracle® VirtualBox.

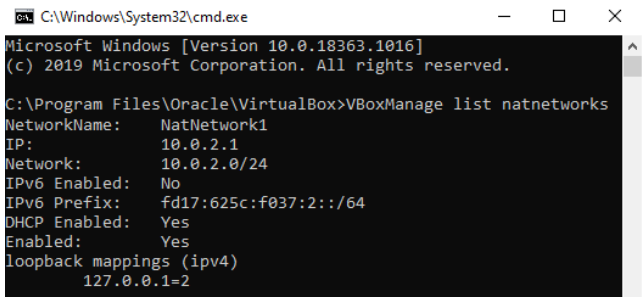

Figure 4: Listing NAT networks.

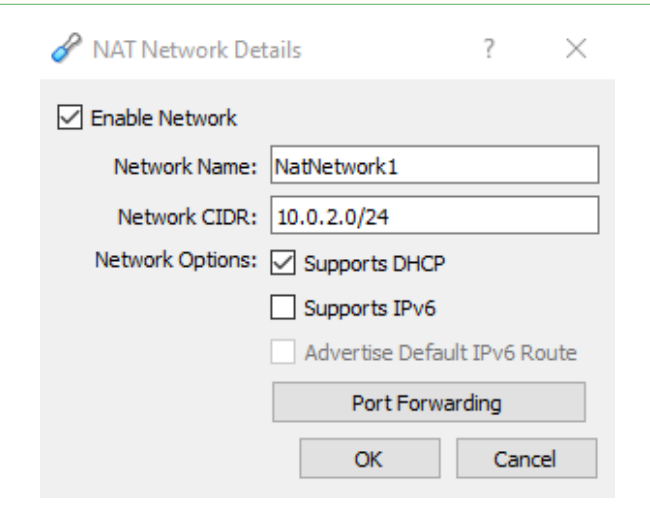

Figure 5: Configuring a NAT network on Oracle® VirtualBox.

Figure 6 below shows the NAT Network formed between the Host (Windows 10) and the three guest virtual machines connected by a virtual switch. The IP addresses were dynamically allocated to the VMs using the dynamic host configuration protocol (DHCP).

The Windows 10 guest was allocated an IP address of 10.0.2.4. The Kali Linux guest was given 10.0.2.6 and the Windows Server 2019 guest got 10.0.2.8. Communication between themselves and the outside world was achieved with such a network. However, the host communication with the host was not achieved as no interface on the host is created to associate with the NAT network.

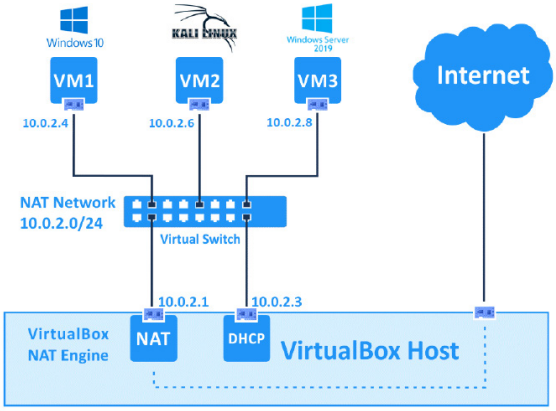

Figure 6: NAT Network formed between hosts and guests.

Host-only network implementation

The host-only network implementation allows each guest virtual machine to communicate with the host and each other. No communication is allowed with the outside world by this technique.

In Oracle® VirtualBox we have created the VirtualBox host-only Ethernet adapter in the host network manager. We have configured the adapter manually. In Windows Control Panel, in the Network and Internet section, under the Network and Sharing Center, we have changed the adapter settings. In the Networking Properties of the VirtualBox Host-Only Ethernet Adapter, we have manually configured the Internet Protocol Version 4 (TCP/IPv4) settings. We set the IPv4 address of our host as 10.0.0.21, the subnet mask of 255.0.0.0 and the default gateway as 10.255.255.255.

Figure 7 shows Oracle® VirtualBox host network manager.

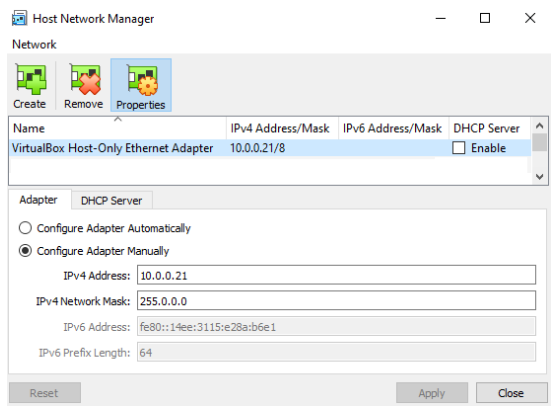

Figure 7: Oracle® VirtualBox host network manager.

We have similarly configured the Internet Protocol Version 4 (TCP/IPv4) settings on the Intel(R) PRO/1000 MT desktop adapter that was virtually created on the Windows 10 guest and the Windows Server 2019. On the Windows 10 guest we have set the IPv4 address as 10.0.0.40, and 10.0.0.50 on the Windows Server 2019 guest. For all the guests, we have set a subnet mask of 255.0.0.0 and a default gateway as 10.255.255.255 (**Figure 8**).

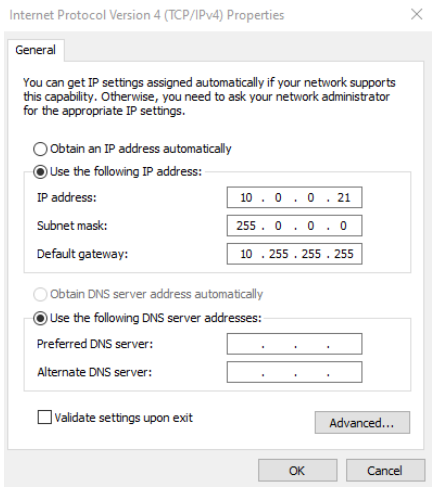

Figure 8: Configuring the IPv4 settings of the host-only Ethernet adapter on the host.

We have customized the Windows defender firewall domain, Private and Public Profiles' Protected network connection properties to uncheck the Ethernet checkboxes. This prevents the firewall from blocking the communication of the virtual machines to the host and to each other (**Figures 9 and 10**).

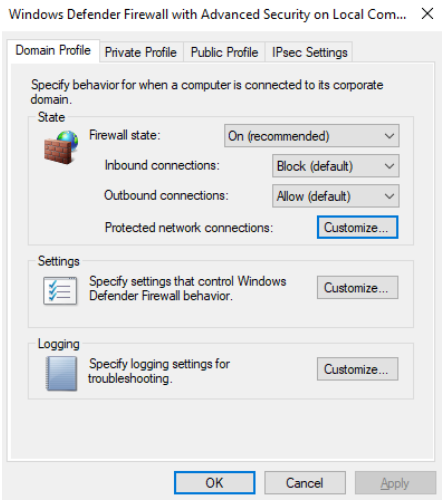

Figure 9: Windows defender firewall properties.

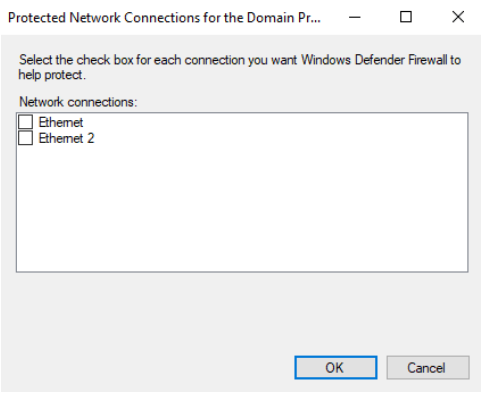

Figure 10: Protected network connections properties.

The Kali Linux guest machine was configured to have a static IP address of 10.0.0.30, with a net mask and default gateway of 255.0.0.0 and 10.255.255.255 respectively, like the Windows guests. The configuration was made by rewriting the file /etc/network/interfaces as shown in **Figure 11** and restarting the networking service with sudo privilege with the command:

sudosystemctl restart networking.service

Static IP configuration in Kali Linux is shown in **Figure 11**.

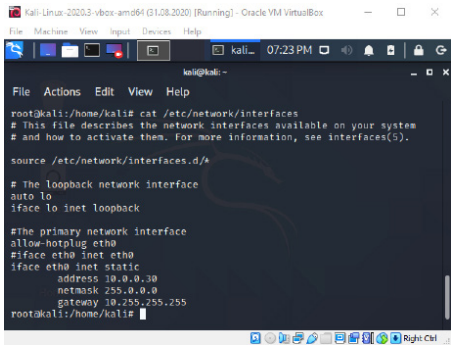

Figure 11: Static IP configuration in Kali Linux.

Figure 12 shows the Host-Only Network formed between the Windows 10 Host and the three guest virtual machines connected by a virtual switch. The IP addresses were configured manually on the virtual machines as shown previously.

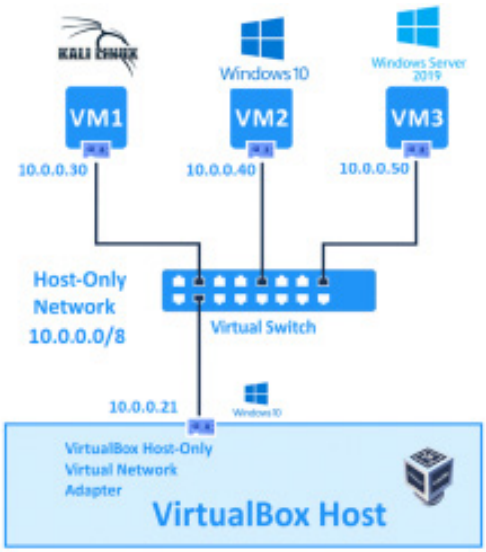

Figure 12: Host-Only network formed between hosts and guests.

Figures 13 and 14 shows the configuration of the Network Settings in Oracle® VirtualBox. The Network Adapter has been enabled. Adapter 1 has been attached to a NAT Network. The network being NAT Network 1 which we created previously. Adapter 2 has been attached to the Host-only Adapter. The adapter is the VirtualBox Host-Only Ethernet Adapter which we created in the Host Network Manager [8]. Host-Only network formed between hosts and guests.

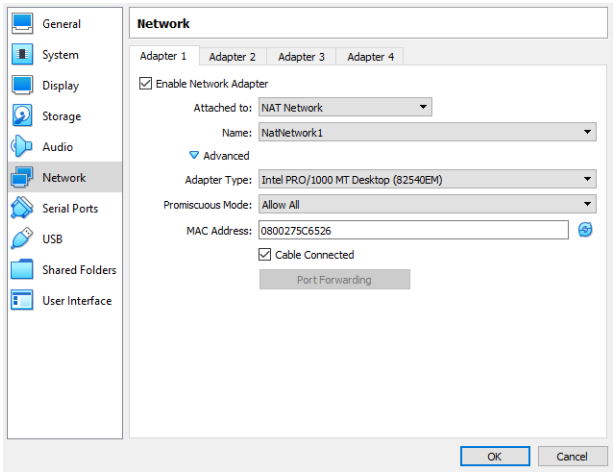

Figure 13: NAT network allotted to adapter 1.

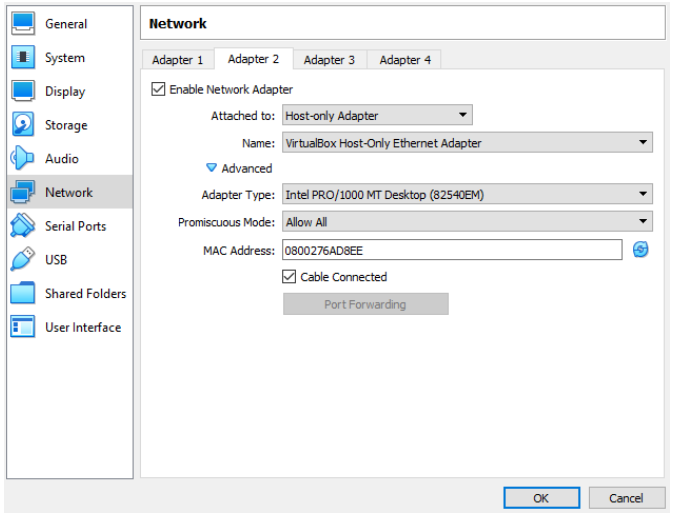

Figure 14: Host-only adapter allotted to adapter 2.

The IP configuration can be verified on Windows machines using the ipconfig command and on Linux machines using the ifconfig command (**Figures 15-18**).

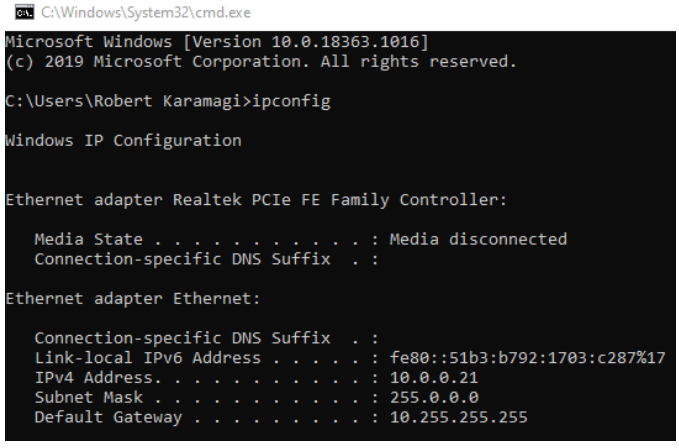

Figure 15: IP configuration of the windows 10 host.

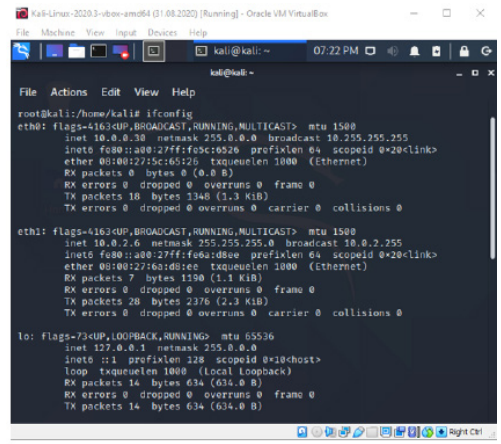

Figure 16: IP configuration of the Kali Linux guest.

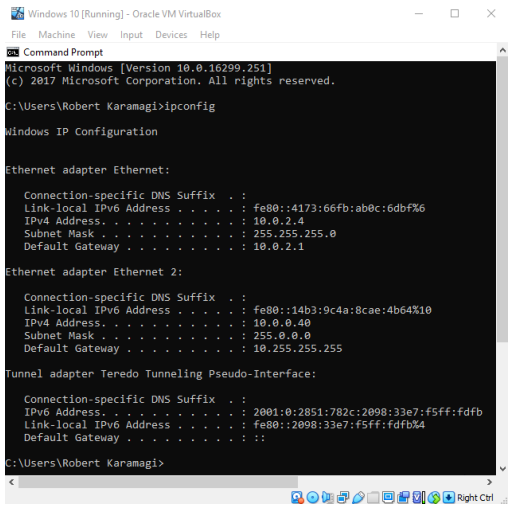

Figure 17: IP configuration of the windows 10 guest.

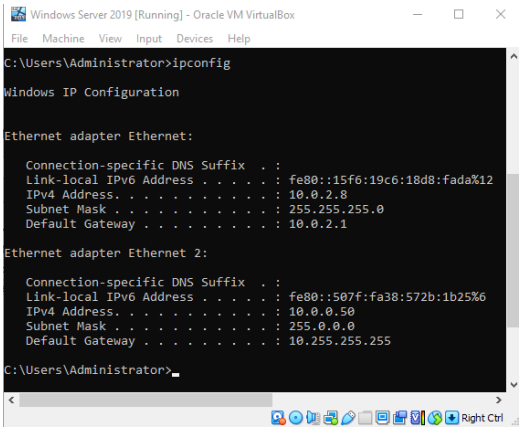

Figure 18: IP configuration of the windows server 2019 guest.

Findings

NAT network communication

Internet connection on the windows 10 host was set and was made available for all the Guest virtual machines using the NAT network service. **Figure 19** shows the mesh network formed between the host and guest virtual machines.

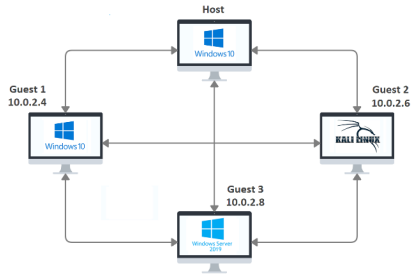

Figure 19: NAT mesh network between host and guest virtual machines.

On the Kali Linux guest machine, using the NAT network, we were able to ping google.com, the Windows 10 guest (10.0.2.4) and the Windows Server 2019 guest (10.0.2.8). **Figure 20** shows pinging www.google.com on the windows 10 host

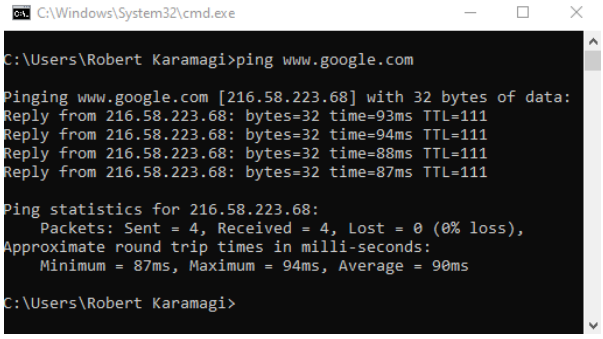

Figure 20: Pinging www.google.com on the windows 10 host.

On the Windows 10 guest machine, using the NAT network, we were able to ping google.com, the Kali Linux guest (10.0.2.6) and the Windows Server 2019 guest (10.0.2.8) which are shown in **Figures 21-23**.

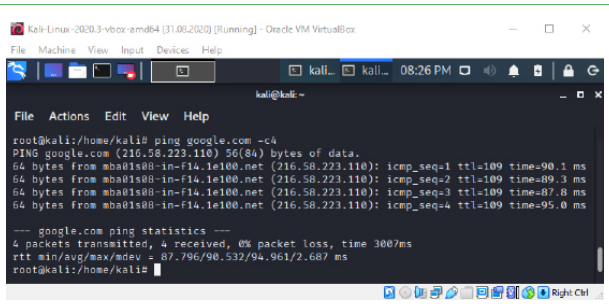

Figure 21: Pinging google.com from the Kali Linux guest.

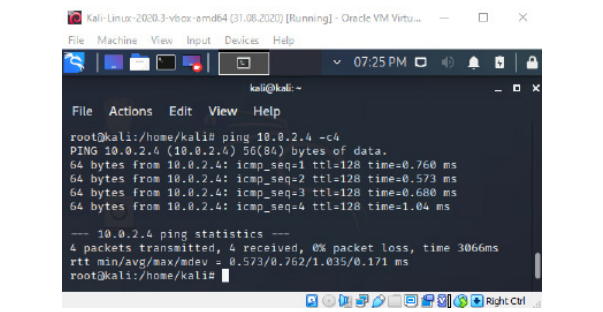

Figure 22: Pinging the windows 10 guest from the Kali Linux guest.

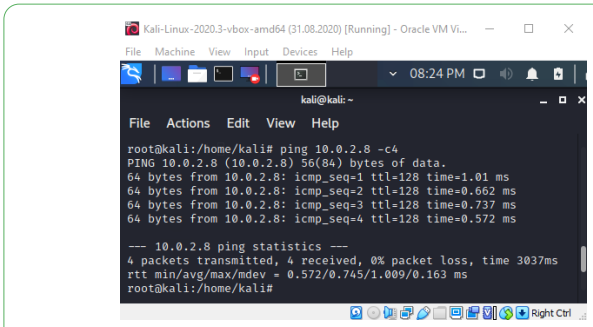

Figure 23: Pinging the windows server 2019 guest from the Kali Linux guest.

network, we were able to ping google.com, the Windows 10 guest (10.0.2.4) and the Kali Linux guest (10.0.2.6) can be seen in **Figures 24-29**.

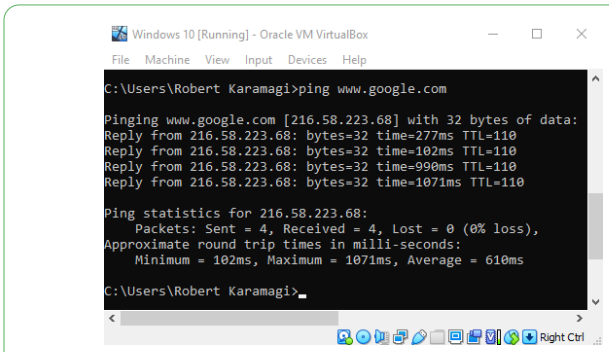

Figure 24: Pinging www.google.com from the windows 10 guest.

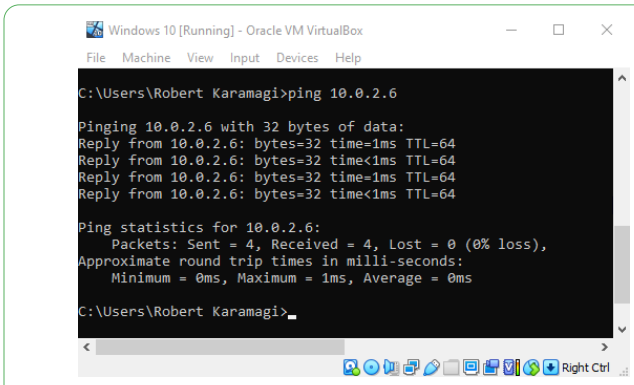

Figure 25: Pinging the Kali Linux guest from the windows 10 guest.

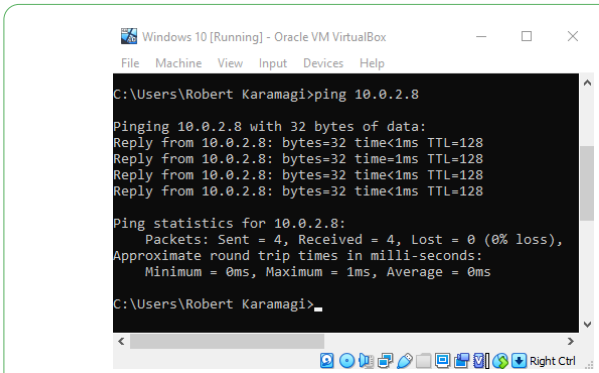

Figure 26: Pinging windows server 2019 guest from the windows 10 guest.

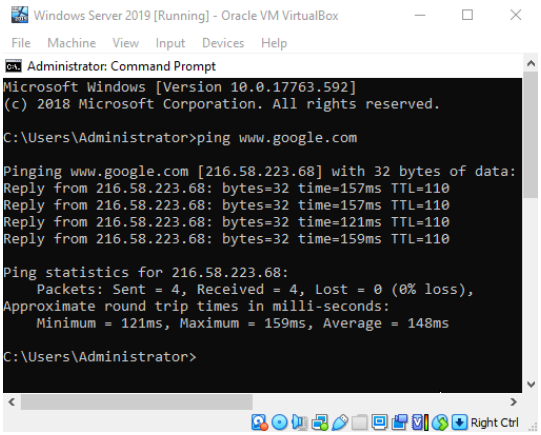

Figure 27: Pinging www.google.com from the windows server 2019 guest.

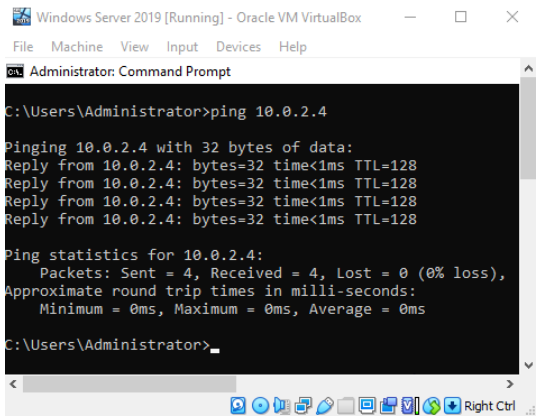

Figure 28: Pinging the windows 10 guest from the windows server 2019 guest.

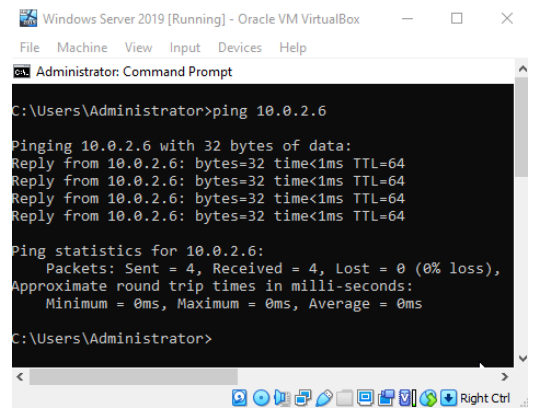

Figure 29: Pinging the Kali Linux guest from the windows server 2019 guest.

Host-only network

Below is a network topology diagram showing the mesh network formed between the Host and Guest virtual machines via the Host-Only networking mode shown in **Figure 30**.

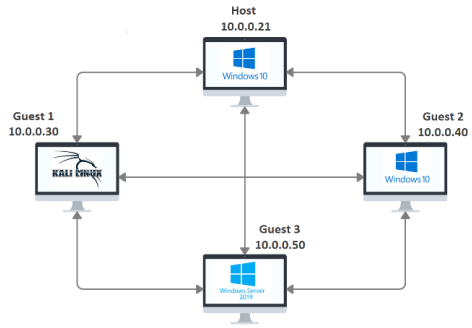

Figure 30: Host-Only mesh network between host and guest virtual machines.

The Windows 10 host, using the Host-Only Ethernet Adapter, was capable of communicating with all the guest virtual machines can be seen in **Figures 31-33**.

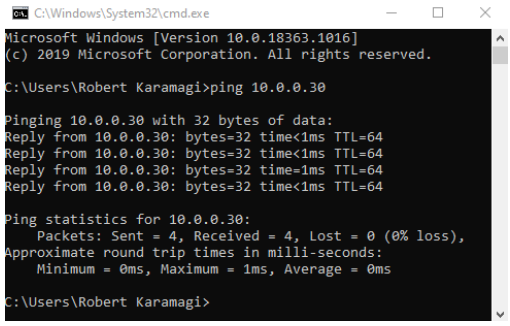

Figure 31: Pinging the Kali Linux guest from the windows 10 host.

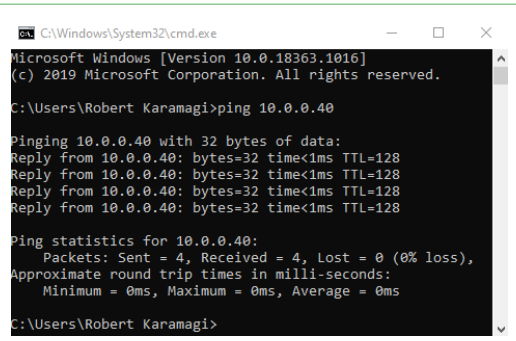

Figure 32: Pinging the windows 10 guest from the windows 10 host.

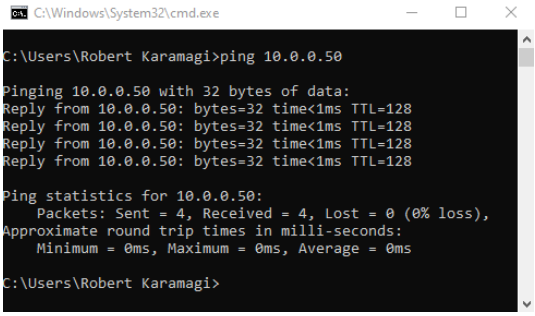

Figure 33: Pinging the windows server 2019 guest from the windows 10 host.

On the Kali Linux guest machine, using the Host-Only network, we were able to ping the Windows 10 Host (10.0.0.21) the Windows 10 guest (10.0.0.40) and the Windows Server 2019 guest (10.0.0.50) can be seen in **Figures 34-36**.

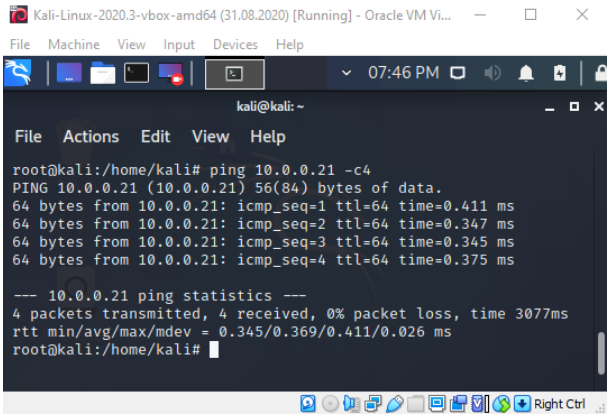

Figure 34: Pinging the windows 10 host from the Kali Linux guest.

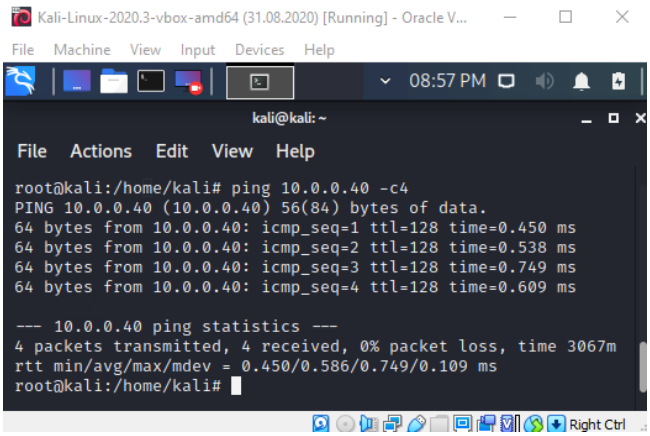

Figure 35: Pinging the windows 10 guest from the Kali Linux guest.

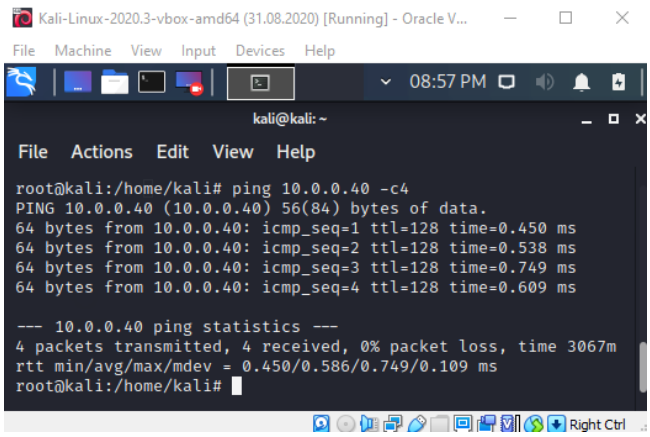

Figure 36: Pinging the windows server 2019 guest from the Kali Linux guest.

On the Windows 10 guest machine, using the Host-Only network, we were able to ping the Windows 10 Host (10.0.0.21) the Kali Linux guest (10.0.0.30) and the Windows Server 2019 guest (10.0.0.50) can be seen in **Figures 37-39**.

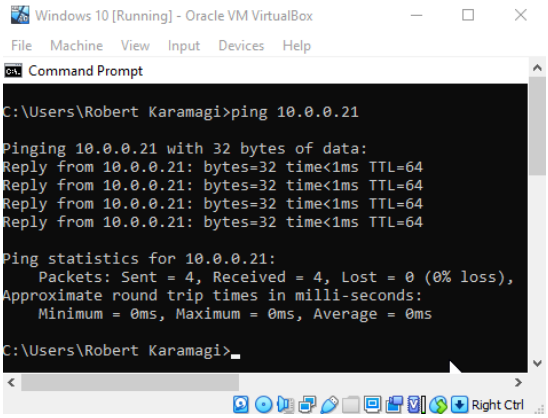

Figure 37: Pinging the windows 10 host from the windows 10 guest.

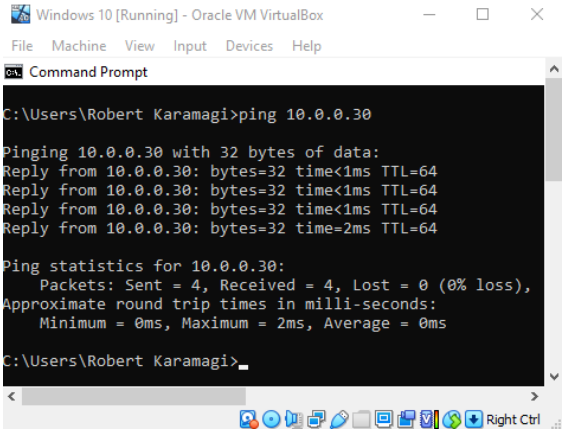

Figure 38: Pinging the Kali Linux guest from the windows 10 guest.

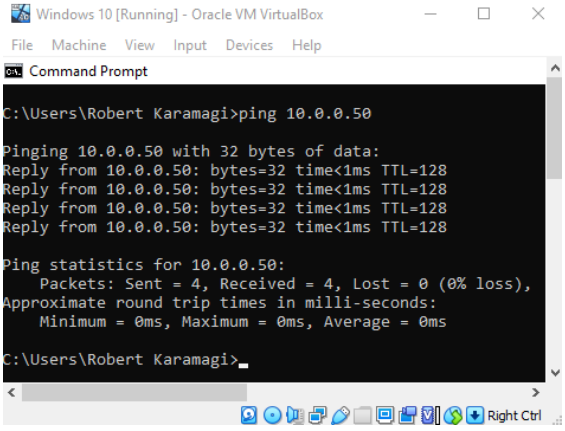

Figure 39: Pinging the windows server 2019 guest from the windows 10 guest.

On the Windows Server 2019 guest machine, using the Host-Only network, we were able to ping the Windows 10 Host (10.0.0.21), the Kali Linux guest (10.0.0.30) and the Windows 10 guest (10.0.0.40) can be seen in **Figures 40-42**.

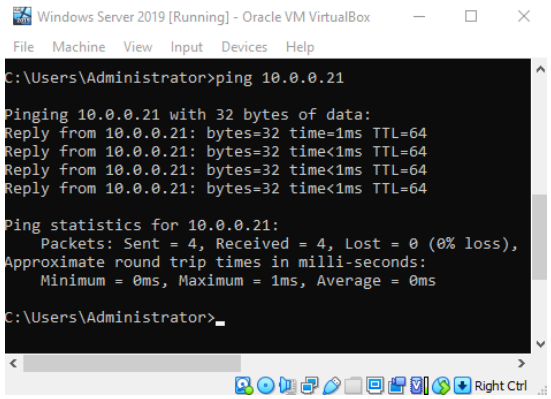

Figure 40: Pinging the windows 10 host from the windows server 2019 guest.

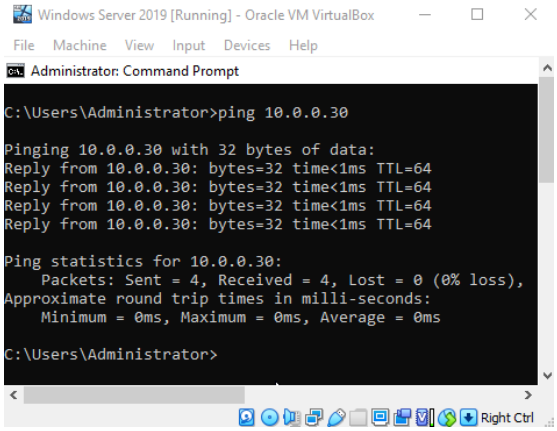

Figure 41: Pinging the Kali Linux guest from the windows server 2019 guest.

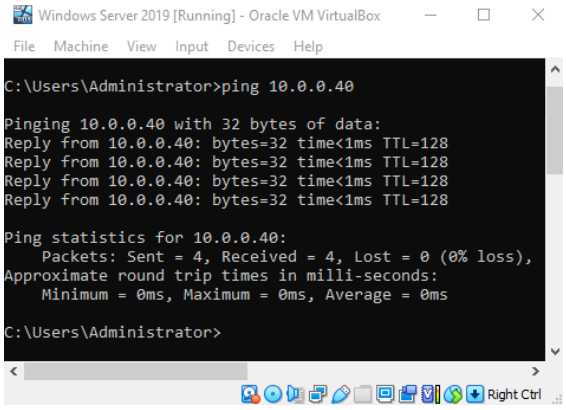

Figure 42: Pinging the windows 10 guest from the windows server 2019 guest.

Results and Discussion

Research Question: What would be our recommendations in configuring the virtual networks in the data center with clients not more than 10,000?

As we can see in **Table 2** a Class B network provides 65,534 host addresses which are way more than the 10,000 needed but shall suit our design criteria.

	Host bits	Host formula	Available hosts
Class A	24-bit	224 - 2	16,777,214
Class B	16-bit	216 - 2	65,534
Class C	8-bit	28 - 2	254

Table 2: Available hosts for various classes.

From a Class B network, we can see from **Table 3** that there are 16,384 available sub networks.

	Subnet mask	Network ID	Network formula	Available networks
Class A	8-bit	1-bit	28-1	128
Class B	16-bit	2-bit	216-2	16,384
Class C	24-bit	3-bit	224-3	2,097,152

Table 3: Available networks for various classes.

Taking a look **Table 4**,which shows the available hosts for a particular Classless Inter-Domain Routing (CIDR) subnet, it is clear that a /18 subnet is our best option as it gives us 16,382 hosts/19 do not give us enough hosts/17 provide too many excess hosts; it is a waste of address space.

CIDR notation	Host formula	Available hosts
/8	2^{32-8} - 2	16,777,214
/9	2^{32-9} - 2	8,388,606
/10	2^{32-10} - 2	4,194,302
/11	2^{32-11} - 2	2,097,150
/12	2^{32-12} - 2	1,048,574
/13	2^{32-13} - 2	524,286
/14	2^{32-14} - 2	262,142
/15	2^{32-15} - 2	131,070
/16	2^{32-16} - 2	65,534
/17	2^{32-17} - 2	32,766
/18	2^{32-18} - 2	16,382
/19	2^{32-19} - 2	8,190
/20	2^{32-20} - 2	4,094
/21	2^{32-21} - 2	2,046
/22	2^{32-22} - 2	1,022
/23	2^{32-23} - 2	510
/24	2^{32-24} - 2	254
/25	2^{32-25} - 2	126
/26	2^{32-26} - 2	62
/27	2^{32-27} - 2	30
/28	2^{32-28} - 2	14
/29	2^{32-29} - 2	6
/30	2^{32-30} - 2	2

Table 4: Available hosts for CIDR subnets.

We can therefore configure the virtual networks in our data center for our 10,000 hosts to be a private internal network i.e. 172.31.0.0/18. As seen in **Table 5**, the subnet mask for our network is 255.255.192.0.

CIDR	Convert to 1s and right pad	Subnet mask
/8	11111111.00000000.00000000.00000000	255.0.0.0
/9	11111111.10000000.00000000.00000000	255.128.0.0
/10	11111111.11000000.00000000.00000000	255.192.0.0
/11	11111111.11100000.00000000.00000000	255.224.0.0
/12	11111111.11110000.00000000.00000000	255.240.0.0
/13	11111111.11111000.00000000.00000000	255.248.0.0
/14	11111111.11111100.00000000.00000000	255.252.0.0
/15	11111111.11111110.00000000.00000000	255.254.0.0
/16	11111111.11111111.00000000.00000000	255.255.0.0
/17	11111111.11111111.10000000.00000000	255.255.128.0
/18	11111111.11111111.11000000.00000000	255.255.192.0
/19	11111111.11111111.11100000.00000000	255.255.224.0
/20	11111111.11111111.11110000.00000000	255.255.240.0
/21	11111111.11111111.11111000.00000000	255.255.248.0
/22	11111111.11111111.11111100.00000000	255.255.252.0
/23	11111111.11111111.11111110.00000000	255.255.254.0
/24	11111111.11111111.11111111.00000000	255.255.255.0
/25	11111111.11111111.11111111.10000000	255.255.255.128
/26	11111111.11111111.11111111.11000000	255.255.255.192
/27	11111111.11111111.11111111.11100000	255.255.255.224
/28	11111111.11111111.11111111.11110000	255.255.255.240
/29	11111111.11111111.11111111.11111000	255.255.255.248
/30	11111111.11111111.11111111.11111100	255.255.255.252

Table 5: CIDR Subnet masks.

For our network, the minimum host address is 172.31.0.1, and the maximum host address is 172.31.63.254. The broadcast address of our network is 172.31.63.255. This network has 16,382 available hosts as from **Table 4**; which is suitable for the virtual network design of our data center with not more than 10,000 hosts.

Conclusion

In our evaluation, we have tested Internet connectivity successfully within all our Virtual Machines, given from our Host. We have also verified the communication between all Virtual Machines with themselves and the host.

References

1. Oracle VM virtualbox (An oracle white paper). VirtualBox.org, 2013.
2. Jayaraman A, Pavankumar R (2012) Comparative study of virtual machine software packages with real operating system. www.semanticscholar.org, 2012.
3. Oracle® VM virtualbox-User manual for release 6.0. docs. oracle.com, 2013.
4. Shah P, Raval V, Nayak A, Ganatra A, Kosta Y (2011) NS2 & networking using desktop virtualization: An application of virtual box. Int J Comput Theory and Eng: 52-57.
5. Chevli C, Youn HY (2005) An efficient VLSI network bridge architecture for local area networks. Int Conf Comput Inf, Adv Comput Inf- ICCI '90: 517-526.
6. Carotenuto L (2016) High performance networking extensions for virtualbox. Anno Accademico 2015-2016.
7. Graniszewski W, Arciszewski A (2016) Performance analysis of selected hypervisors (Virtual Machine Monitors-VMMs). Int J Electron Telecomm 62: 231-236.
8. Oracle VM 3 : Building a demo environment using Oracle VM virtualbox. www.oracle.com, 2012.