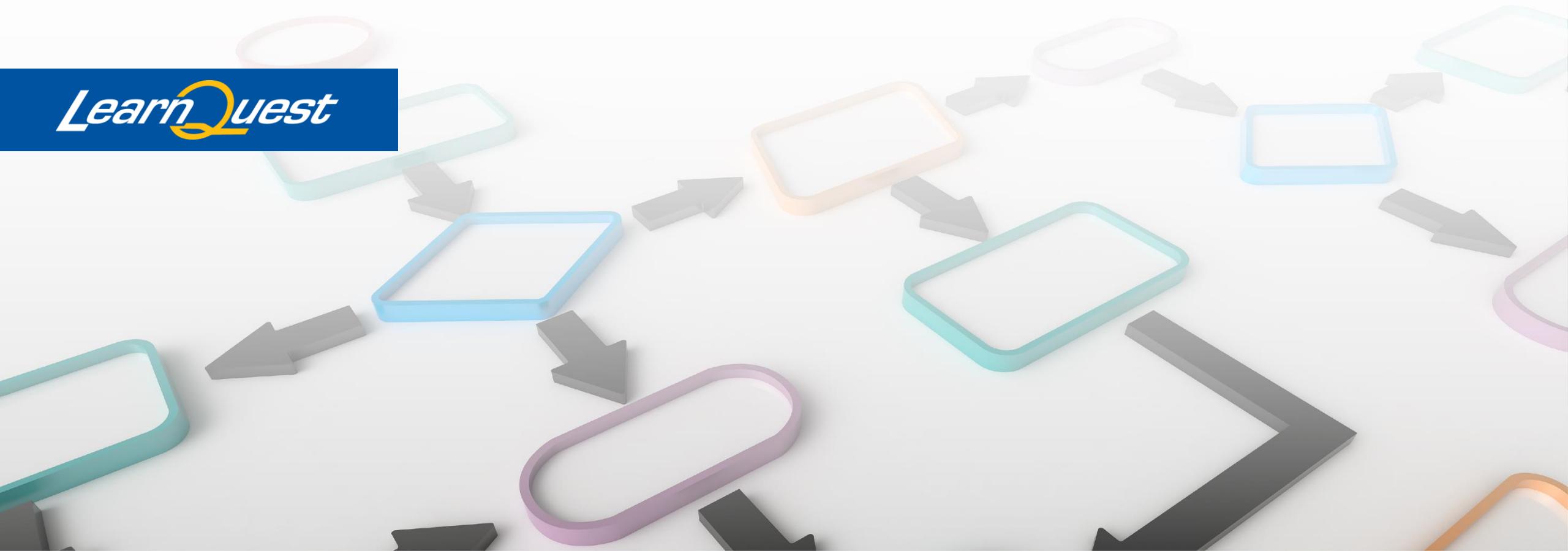


Linux Cloud and DevOps

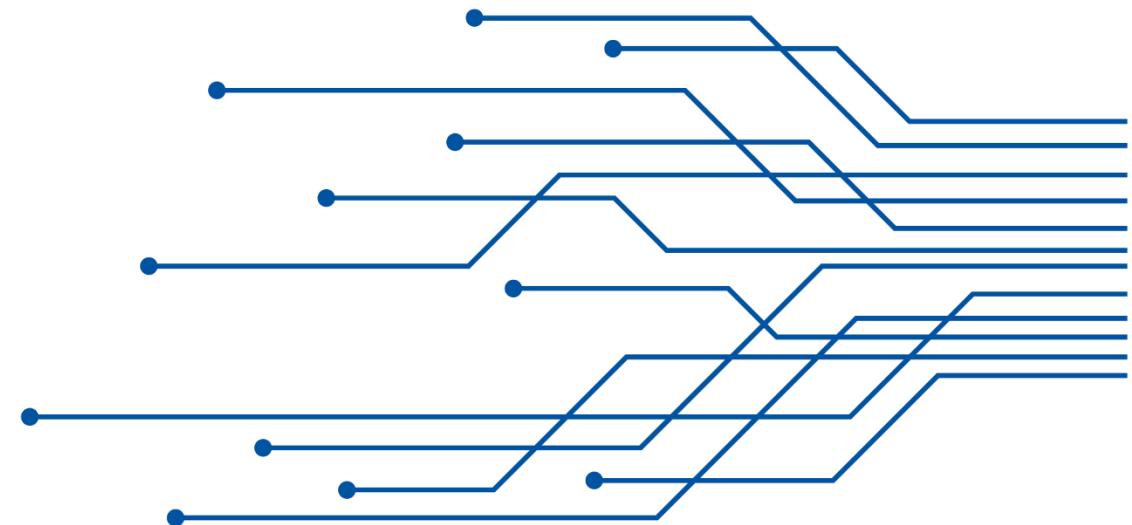
- 4th Course in Linux Foundations Specialization



Virtualization

In this module, we look at how to virtualize guest operating systems running under a Linux host operating system.

2



Learning Objectives

Virtualization

Upon completion of this module, learners will be able to:

- Differentiate between Hypervisor Types
- Manage Virtual Machines in Linux
- Describe Bootstrapping
- Virtualize Network Connections

Lesson 1

Hypervisor Types

In this lesson we look at Hypervisor Types

Type I Hypervisors

Commonly called bare-metal hypervisors

The hypervisor system runs directly on the server hardware, with no middleman

The hypervisor software interacts directly with the CPU, memory, and storage on the system, allocating them to each virtual machine as needed

Examples:

- Xen Server
- KVM
- VMware ESXi
- Microsoft Hyper-V

Linux Type I Hypervisors

KVM - Linux Kernel-based Virtual Machine uses a standard Linux kernel along with a special hypervisor module. Can host any type of guest operating systems.

XEN - XEN Project is an open-source project for hardware virtualization. Includes additional software besides the hypervisor software, for managing the hypervisor from a guest operating system.

Type II Hypervisors



Commonly called hosted hypervisors because they run on top of an existing operating system



The hypervisor software runs like any other application on the host operating system



Examples:

VirtualBox
VMware WorkStation

Type II Host and/or Guest OS

Be

Linux can be either the Host or the Guest OS in Type II hypervisors

Run

Linux can Run as a guest with another operating system as the host

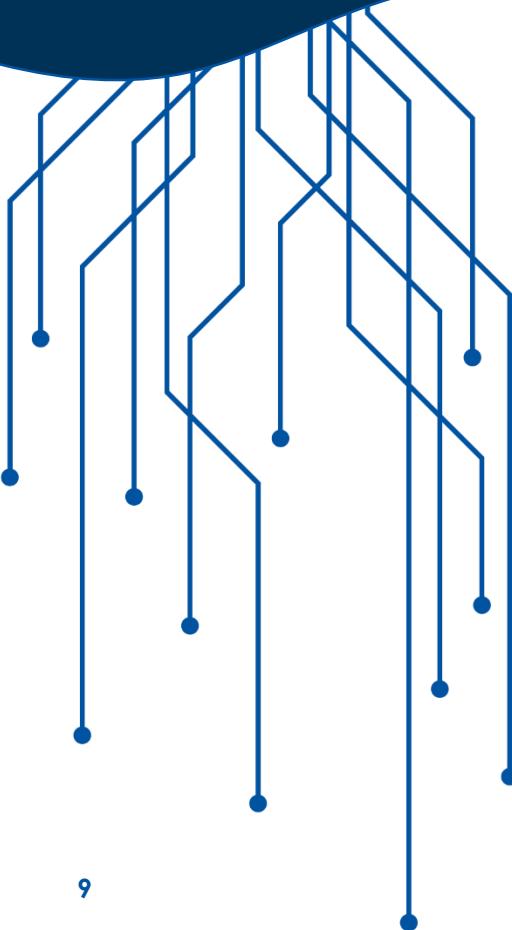
Run

Linux can Run as the host with another operating system as the guest

Be

Linux can be both the host and the guest OS in Type II hypervisors

Lesson 1 Review



Type 1 Hypervisors run on the hardware



Type II Hypervisors run on an installed OS



VirtualBox is the Type II Hypervisor that most use on Linux

Lesson 2

Virtual Machine Tools

In this lesson we look at Virtual Machine Tools

Libvirt Toolkit



An application programming interface (API) library that is incorporated into several open-source VMMs (hypervisors), such as KVM

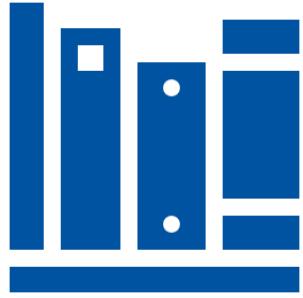


A daemon, libvirtd, that operates on the VM host system and executes any needed VM guest system management tasks, such as starting and stopping the VM

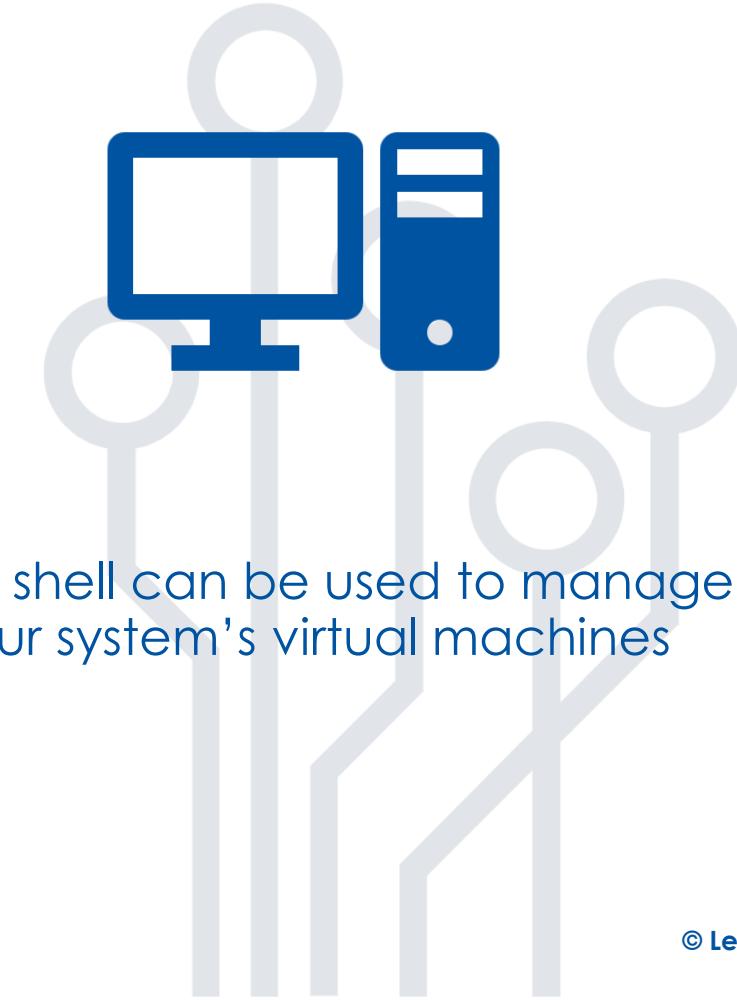


Command-line utilities, such as virt-install and virsh, that operate on the VM host system and are used to control and manage VM guest systems

Virsh



Virsh shell uses the libvirt library



Basic shell can be used to manage
your system's virtual machines

Virsh Commands

Command	Description
help	Prints basic help information.
list	Lists all guests.
dumpxml	Outputs the XML configuration file for the guest.
create	Creates a guest from an XML configuration file and starts the new guest.
start	Starts an inactive guest.
destroy	Forces a guest to stop.
define	Outputs an XML configuration file for a guest.
domid	Displays the guest's ID.
domuuid	Displays the guest's UUID.
dominfo	Displays guest information.
domname	Displays the guest's name.
domstate	Displays the state of a guest.
quit	Quits the interactive terminal.
reboot	Reboots a guest.
restore	Restores a previously saved guest stored in a file.
resume	Resumes a paused guest.
save	Saves the present state of a guest to a file.
shutdown	Gracefully shuts down a guest.
suspend	Pauses a guest.
undefine	Deletes all files associated with a guest.
migrate	Migrates a guest to another host.

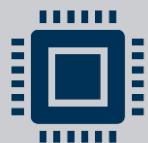
Virtual Machine Manager (VMM)



A lightweight desktop application for creating and managing virtual machines.

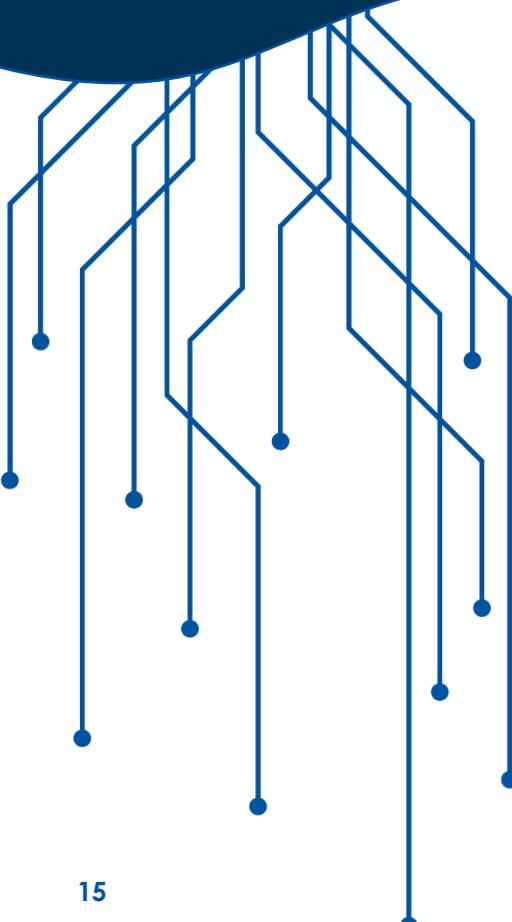


A Python script that displays a GUI and is part of the virt-manager package.



The Virtual Machine Manager can be started from a terminal emulator within the graphical environment via the `virt-manager` command

Lesson 2 Review



Libvirt works with many Linux hypervisors



Virsh uses Libvirt to create a shell to manage hypervisors



VMM is a graphical tool to create and manage VMs

Lesson 3

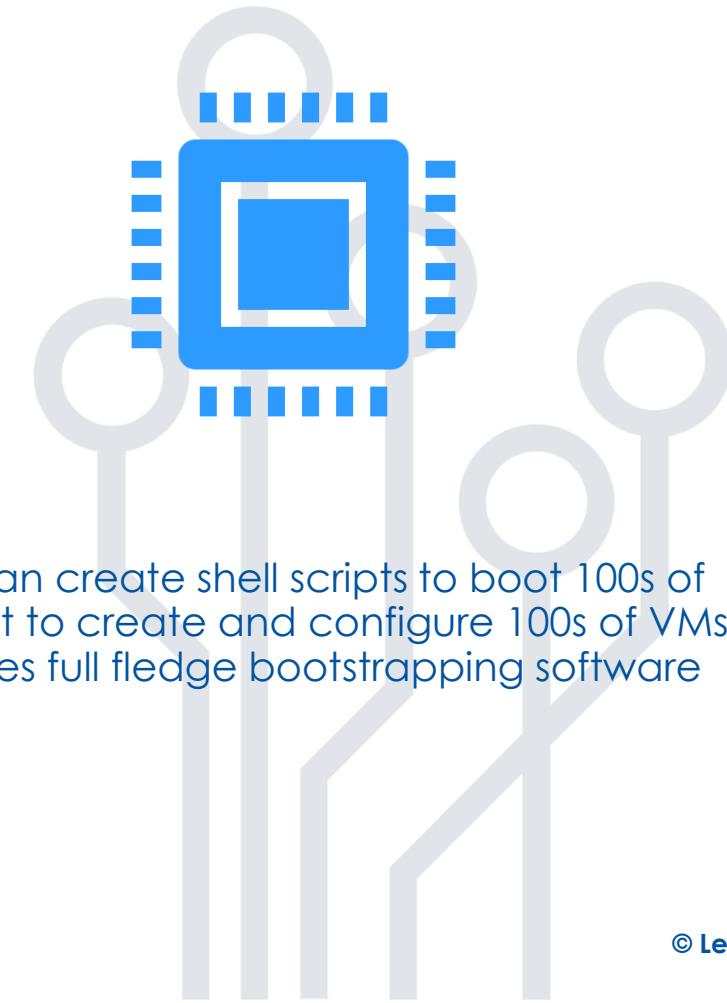
Bootstrapping

In this lesson we look at
Bootstrapping VMs

What is Bootstrapping?



Starting a few VMs from a GUI is easy



You can create shell scripts to boot 100s of VMs but to create and configure 100s of VMs requires full fledged bootstrapping software

Kickstart

Red Hat created the kickstart installation method to automate the process of creating VMs

A system administrator can create a single file containing the answers to all the questions that would normally be asked during a typical installation

Kickstart files can be kept on a server system and read by individual computers during the installation

This installation method can support the use of a single kickstart file to install Linux on multiple machines

Cloud-init

Canonical product

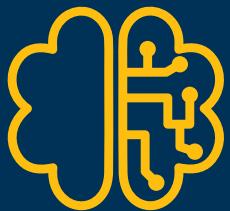
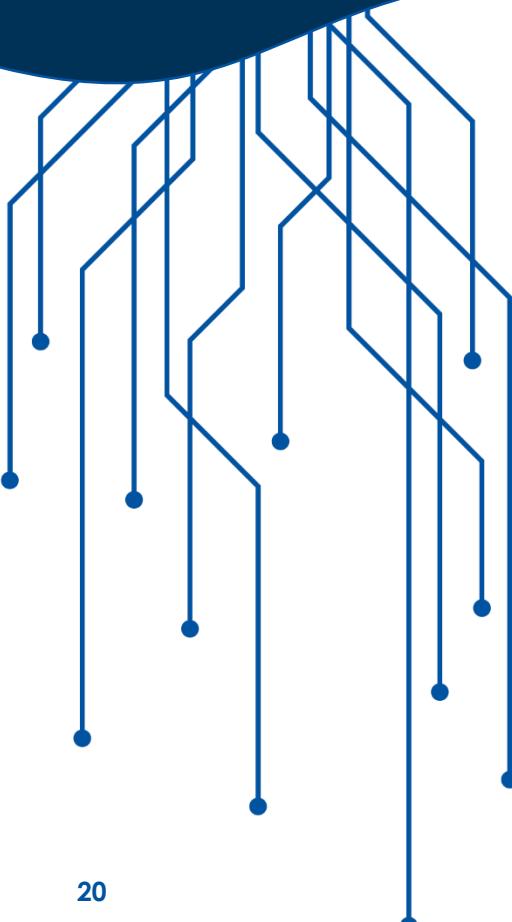
Used to create modified VMs from cloned VMs

Supported by most Linux distributions

Can employ pregenerated OpenSSH private keys
for access

Customization scripts can be used for bootstrapping via user-data in YAML
files or via command line

Lesson 3 Review



Bootstrapping is designed to automate the creation and configuration of VMs



Kickstart uses centralized network file to answer installation questions



Cloud-init can install private keys that can be used to connect to the VMs via SSH

Lesson 4

Virtual Networks

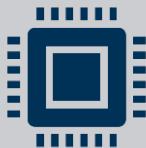
In this lesson we look at Virtual Networks



Virtualized NICs and Switches



Applications on physical systems can reach the outside world via a network interface card (NIC) and a connected network.



Virtualized machines can have virtualized NICs and virtualized internal switches

Network Virtualization Concepts



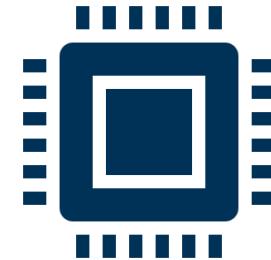
VLAN

LAN that can be physically located in a local geographic area

Based on logical and virtual connections

Uses layer 2 to broadcast messages

Router uses layer 3 to implement the VLAN



Overlay network

Network virtualization method that uses encapsulation & communication bandwidth tunneling

Employs virtual switches, tunneling protocols, and SDN

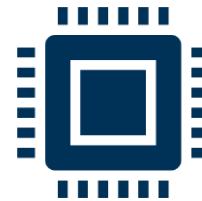
Three Virtual Network Types



Host-only adapter

Connects to a virtual network contained within VM's host system

No connection to external physical (or virtual) network to which host is attached



Bridged adapter

VM is a node on LAN or VLAN to which host system is attached

VM gets its own IP address and can be seen on network
Virtual NIC is connected to a host machine's physical NIC



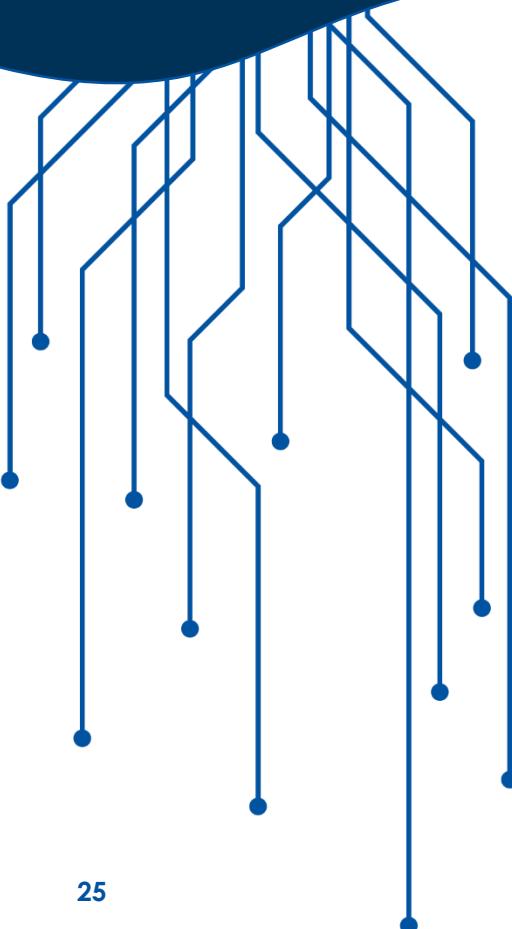
NAT adapter

NAT table is maintained by hypervisor instead of network device

IP address of host is employed as single IP address that is sent out onto external network

Each VM has its own IP address within host system's virtual network

Lesson 4 Review



A host-only adapter cannot see beyond the host OS



A bridged adapter gets its own IP on the network



A NAT adapter has a private IP address beyond the host OS