KVM Virtualization on Linux

Tom Dean
Linux Training Architect

About the Course

In this lesson, we are going to preview the **KVM Virtualization on Linux** course. We will talk about the scope of the course and what skills and experience you should possess. When you finish this lesson, you should have a good understanding of what the course is about.

In this course, we're going to explore **KVM Virtualization on Linux** from **four** perspectives:

Workstation virtualization using **CentOS 8**

- Virtual Machine Manager (virt-manager)

Web-based administration using **CentOS 8**

- Cockpit/Web Console

CLI administration using **CentOS 8**

Exploring oVirt on **CentOS 7**

In this course, we're going to explore **KVM Virtualization on Linux** from **four** perspectives:

Workstation virtualization using **CentOS 8**

- Virtual Machine Manager (virt-manager)

Web-based administration using **CentOS 8**

- Cockpit/Web Console

CLI administration using **CentOS 8**

Exploring oVirt on **CentOS 7**

In this course, we're going to explore **KVM Virtualization on Linux** from **four** perspectives:

Workstation virtualization using **CentOS 8**

- Virtual Machine Manager (virt-manager)

Web-based administration using **CentOS 8**

- Cockpit/Web Console

CLI administration using **CentOS 8**

Exploring oVirt on **CentOS 7**

In this course, we're going to explore **KVM Virtualization on Linux** from **four** perspectives:

Workstation virtualization using **CentOS 8**

- Virtual Machine Manager (virt-manager)

Web-based administration using **CentOS 8**

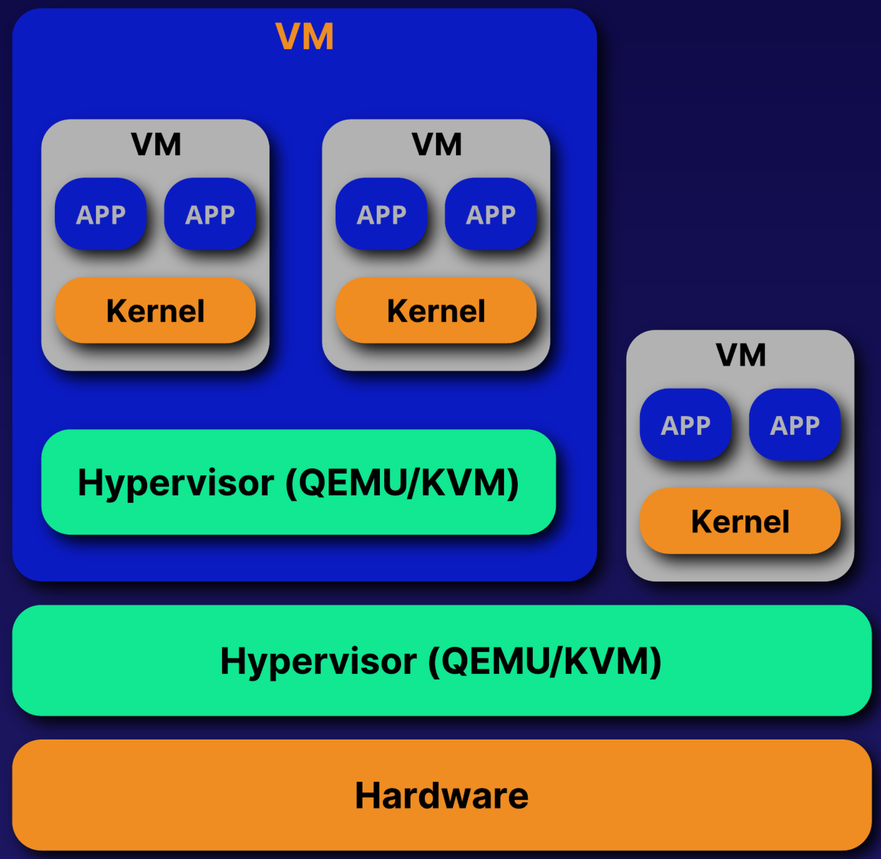
- Cockpit/Web Console

CLI administration using **CentOS 8**

Exploring oVirt on **CentOS 7**

Nested Virtualization

Due to the course requirement for nested virtualization, there are no **hands-on labs**, as there is no support for nested virtualization in hands-on labs. For the same reason, this will not work in the **Cloud Playground**.
Therefore, all lessons are follow-along, using your own computer (**optional** but **recommended**).



Course Prerequisites

What You Should Be Experienced/Proficient With

1 Linux

- Command line
- GUI
- Networking
- Storage

2 RHEL Distributions

- **Fedora**/**CentOS**/**RHEL**
  - **dnf**/**yum** package management

3 Hardware

- **Intel** CPU (with **VT-x**)
- **AMD** CPU (with **AMD-V**)

4 Other

- Virtualization experience

What You Should Be Experienced/Proficient With

1 **Linux**

- Command line
- GUI
- Networking
- Storage

2 **RHEL Distributions**

- **Fedora**/**CentOS**/**RHEL**
  - **dnf**/**yum** package management

3 **Hardware**

- **Intel** CPU (with **VT-x**)
- **AMD** CPU (with **AMD-V**)

4 **Other**

- Virtualization experience

What You Should Be Experienced/Proficient With

1 Linux

- Command line
- GUI
- Networking
- Storage

2 RHEL Distributions

- **Fedora**/**CentOS**/**RHEL**
  - **dnf**/**yum** package management

3 Hardware

- **Intel** CPU (with **VT-x**)
- **AMD** CPU (with **AMD-V**)

4 Other

- Virtualization experience

What You Should Be Experienced/Proficient With

1 Linux

- Command line
- GUI
- Networking
- Storage

2 RHEL Distributions

- **Fedora**/**CentOS**/**RHEL**
  - **dnf**/**yum** package management

3 Hardware

- **Intel** CPU (with **VT-x**)
- **AMD** CPU (with **AMD-V**)

4 Other

- Virtualization experience

CentOS

redhat

(Optional) Hardware Requirements

Hardware/Software Requirements for Following Along

1 Computer — Bare Metal

- Intel CPU with **VT-x**
- AMD CPU with **AMD-V**
- Plenty of RAM
- The more cores, the better
- Reasonable disk space

2 Alternative

- Hypervisor
  - Nested virtualization support
  - Self-supported
  - Lessons to assist with setup

Hardware/Software Requirements for Following Along

1 **Computer — Bare Metal**

- Intel CPU with **VT-x**
- AMD CPU with **AMD-V**
- Plenty of RAM
- The more cores, the better
- Reasonable disk space

2 **Alternative**

- Hypervisor
  - Nested virtualization support
  - Self-supported
  - Lessons to assist with setup

Hardware

I'm using:

- Two **Dell Precision T3500** workstations
  - Single **Intel 6-core Xeon** (each)
  - **24GB** RAM (each)
  - **RAID-1** 1TB HDD
    - **Additional** 3TB HDD
  - **Two** network connections:
    - “**Public**” network
    - “**Private**” network

CentOS 7 / CentOS 8

Be Patient

Virtualization can mean installs **take time**!

- Get up and **stretch**
- Get a **snack**
- **Review** past lessons
- Get **refreshed** for when action resumes

We hope you enjoy the course!

In this course, we've aspired to provide an interesting journey through four scenarios that use **KVM Virtualization on Linux**.

Tom Dean
Linux Training Architect

About the Training Architect

Interests

- Quality Time with **Kids**
- **Boating**
- Obsolete **Electronics**
- **Music**
- **Tinkering** in General
- **Mentoring/Teaching**

Started on the Apple II in the early 1980s

Purdue University

25 years experience in Information Technology

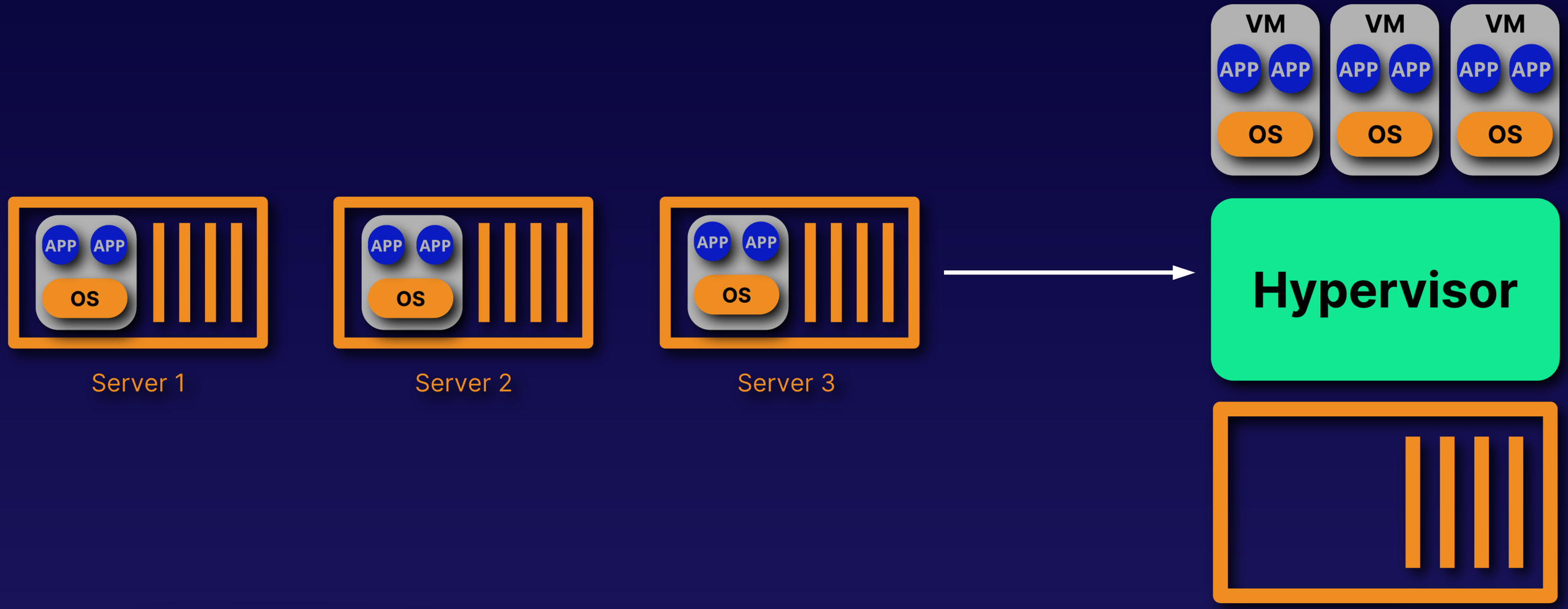
Instructor:

Tom Dean

LINUX TRAINING ARCHITECT

Thanks for getting to know me!

What Is Virtualization?



Virtualization allows you to decouple your services and applications from the physical (server) layer and deploy them more efficiently and reliably than the traditional 1:1 service/application-to-physical server relationship.

History of Virtualization

Mainframes — The 1960s

MIT announces Project MAC (Mathematics and Computation/Multiple Access Computer):

- $2 million grant from DARPA
- IBM passes
- GE becomes *the* vendor of choice

Mainframes — The 1960s

Bell Labs also has a need for *multi-user* system

- IBM designs **CP-40** mainframe:
  - Evolves into **CP-67**
  - **CP**/**CMS** (Control Program/Console Monitor System):
    - **1972** — first *stable* release
    - **CP** runs on the mainframe:
      - Creates the **VMs**
      - **VMs** run the **CMS**:
        - User interacts with *these*
        - Each user gets their own operating system:
          - *More stable and secure*

Hardware Virtualization — Late ’80s and Beyond

Insignia Solutions

SoftPC:

UNIX — 1987:
Run DOS applications on UNIX Workstation

Mac — 1989:
Could run DOS and Windows applications

Apple/Connectix

Virtual PC — 1997:
Run Windows on Mac

VMWare

VMWare Workstation — 1999:
Initially ran on Windows

ESX/GSX Server — 2001:

ESX Server:
Type-1 hypervisor

GSX Server:
Type-2 hypervisor

Hardware Virtualization — Late ’80s and Beyond

Insignia Solutions

SoftPC:

UNIX — 1987:
Run DOS applications on UNIX Workstation

Mac — 1989:
Could run DOS and Windows applications

Apple/Connectix

Virtual PC — 1997:
Run Windows on Mac

VMWare

VMWare Workstation — 1999:
Initially ran on Windows

ESX/GSX Server — 2001:

ESX Server:
Type-1 hypervisor

GSX Server:
Type-2 hypervisor

Hardware Virtualization — Late ’80s and Beyond

Insignia Solutions

SoftPC:

UNIX — 1987:
Run DOS applications on UNIX Workstation

Mac — 1989:
Could run DOS and Windows applications

Apple/Connectix

Virtual PC — 1997:
Run Windows on Mac

VMWare

VMWare Workstation — 1999:
Initially ran on Windows

ESX/GSX Server — 2001:

ESX Server:
Type-1 hypervisor

GSX Server:
Type-2 hypervisor

Java Virtual Machine — The Early 1990s

1. **Sun Microsystems began Project Stealth in 1990**
   - In 1995, the project officially became Java:
     - Targeted toward WWW
     - ***Write application once, run anywhere:***
       - Java Runtime Environment (JRE)
       - Compiles application, just in time (JIT)
       - Developer doesn't have to worry about the operating system:
         - *Just write the code*

2. **Java Virtual Machine**
   - Small operating system, a “virtual machine”
   - Runs the Java application

Sun microsystems

Java Virtual Machine — The Early 1990s

1 Sun Microsystems began Project Stealth in 1990

- In 1995, the project officially became Java:
  - Targeted toward WWW
  - ***Write application once, run anywhere:***
    - Java Runtime Environment (JRE)
    - Compiles application, just in time (JIT)
    - Developer doesn't have to worry about the operating system:
      - *Just write the code*

2 Java Virtual Machine

- Small operating system, a “virtual machine”
- Runs the Java application

Sun microsystems

Java Virtual Machine — The Early 1990s

1 Sun Microsystems began Project Stealth in 1990

- In 1995, the project officially became Java:
  - Targeted toward WWW
  - ***Write application once, run anywhere:***
    - Java Runtime Environment (JRE)
    - Compiles application, just in time (JIT)
    - Developer doesn't have to worry about the operating system:
      - *Just write the code*

2 Java Virtual Machine

- Small operating system, a “virtual machine”
- Runs the Java application

Sun microsystems

Java Virtual Machine — The Early 1990s

1 Sun Microsystems began Project Stealth in 1990

- In 1995, the project officially became Java:
  - Targeted toward WWW
  - ***Write application once, run anywhere:***
    - Java Runtime Environment (JRE)
    - Compiles application, just in time (JIT)
    - Developer doesn't have to worry about the operating system:
      - *Just write the code*

2 Java Virtual Machine

- Small operating system, a “virtual machine”
- Runs the Java application

Hardware-Assisted Virtualization — 2005/2006

In short, hardware virtualization is where we add features to the CPU that assist and accelerate virtualization by *moving* these functions from the software (hypervisor) to the hardware (CPU).

1 Intel Virtualization (VT-x)

- CPU flag for **VT-x** capability is "**VMX**"
- **`grep -i vmx /proc/cpuinfo`**
- "**VMX**" stands for **Virtual Machine Extensions**:
  - Adds *13 new instructions*
  - Guest OS perceives itself as running with full privilege:
    - Host os remains protected
  - Extended Page Tables (**EPT**):
    - Intel technology for the Memory Management Unit (**MMU**)

2 AMD Virtualization (AMD-V)

- CPU flag for **AMD-V** capability is "**SVM**":
  - AMD **Secure Virtual Machine**, later **AMD-V**
  - **`grep -i svm /proc/cpuinfo`**
- Rapid Virtualization Indexing (**RVI**):
  - AMD technology for the processor Memory Management Unit (**MMU**)

Hardware-Assisted Virtualization — 2005/2006

In short, hardware virtualization is where we add features to the CPU that assist and accelerate virtualization by *moving* these functions from the software (hypervisor) to the hardware (CPU).

1 Intel Virtualization (VT-x)

- CPU flag for **VT-x** capability is “**VMX**”
- **`grep -i vmx /proc/cpuinfo`**
- “**VMX**” stands for **Virtual Machine Extensions**:
  - Adds *13 new instructions*
  - Guest OS perceives itself as running with full privilege:
    - Host os remains protected
  - Extended Page Tables (**EPT**):
    - Intel technology for the Memory Management Unit (**MMU**)

2 AMD Virtualization (AMD-V)

- CPU flag for **AMD-V** capability is “**SVM**”:
  - AMD **Secure Virtual Machine**, later **AMD-V**
  - **`grep -i svm /proc/cpuinfo`**
- Rapid Virtualization Indexing (**RVI**):
  - AMD technology for the processor Memory Management Unit (**MMU**)

Hardware-Assisted Virtualization — 2005/2006

In short, hardware virtualization is where we add features to the CPU that assist and accelerate virtualization by *moving* these functions from the software (hypervisor) to the hardware (CPU).

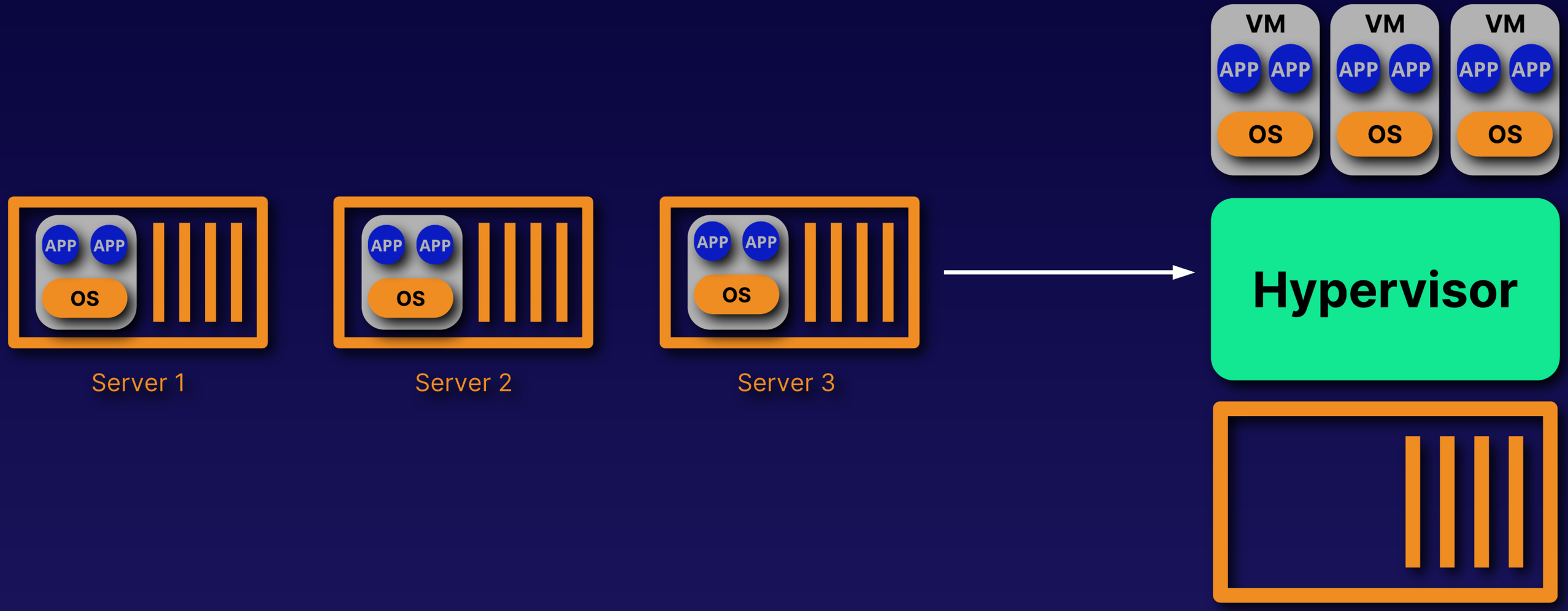
1 Intel Virtualization (VT-x)

- CPU flag for **VT-x** capability is "**VMX**"
- **`grep -i vmx /proc/cpuinfo`**
- "**VMX**" stands for **Virtual Machine Extensions**:
  - Adds *13 new instructions*
  - Guest OS perceives itself as running with full privilege:
    - Host os remains protected
  - Extended Page Tables (**EPT**):
    - Intel technology for the Memory Management Unit (**MMU**)

2 AMD Virtualization (AMD-V)

- CPU flag for **AMD-V** capability is "**SVM**":
  - AMD **Secure Virtual Machine**, later **AMD-V**
  - **`grep -i svm /proc/cpuinfo`**
- Rapid Virtualization Indexing (**RVI**):
  - AMD technology for the processor Memory Management Unit (**MMU**)

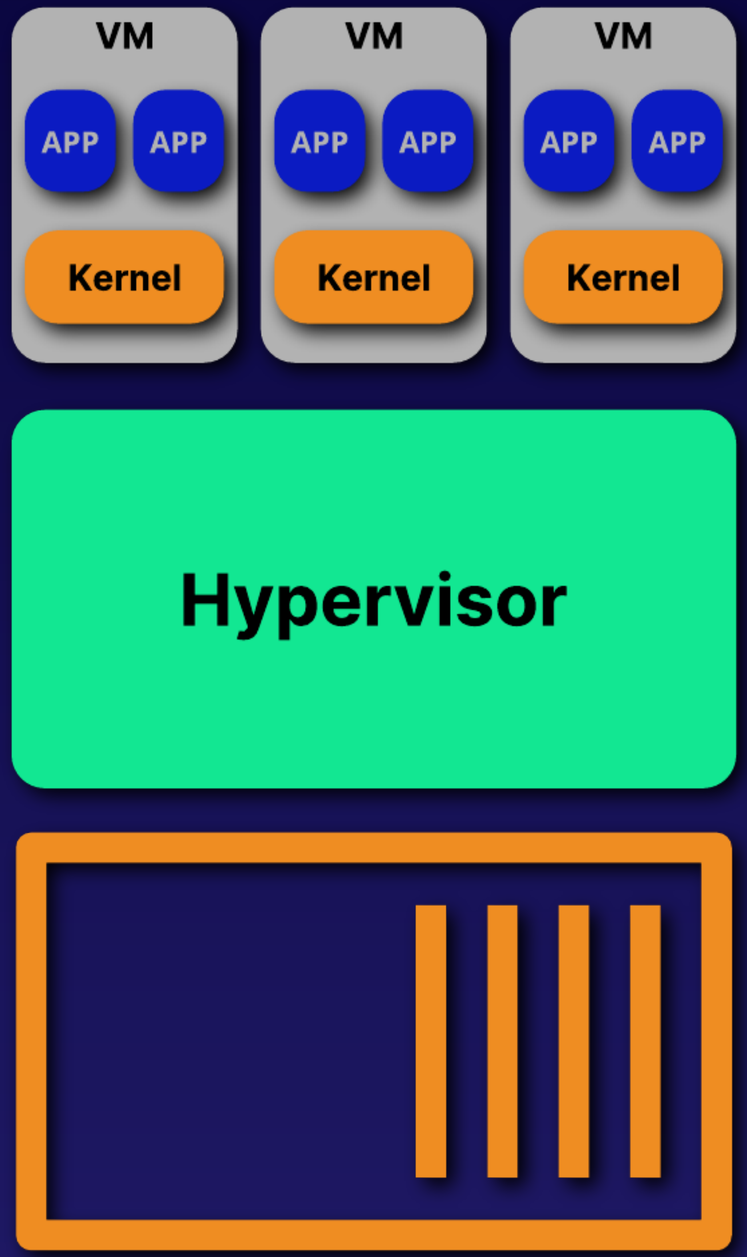
How Does Virtualization Work?



Virtualization creates a **virtual machine** for the guest operating system to live within, with *virtual resources* instead of *physical resources*. Physical resources can also be mapped to a guest where necessary.

What Is a Hypervisor?

A **hypervisor** is software that sits between the guest (virtual machine) operating system and either the hardware itself or another operating system (host).

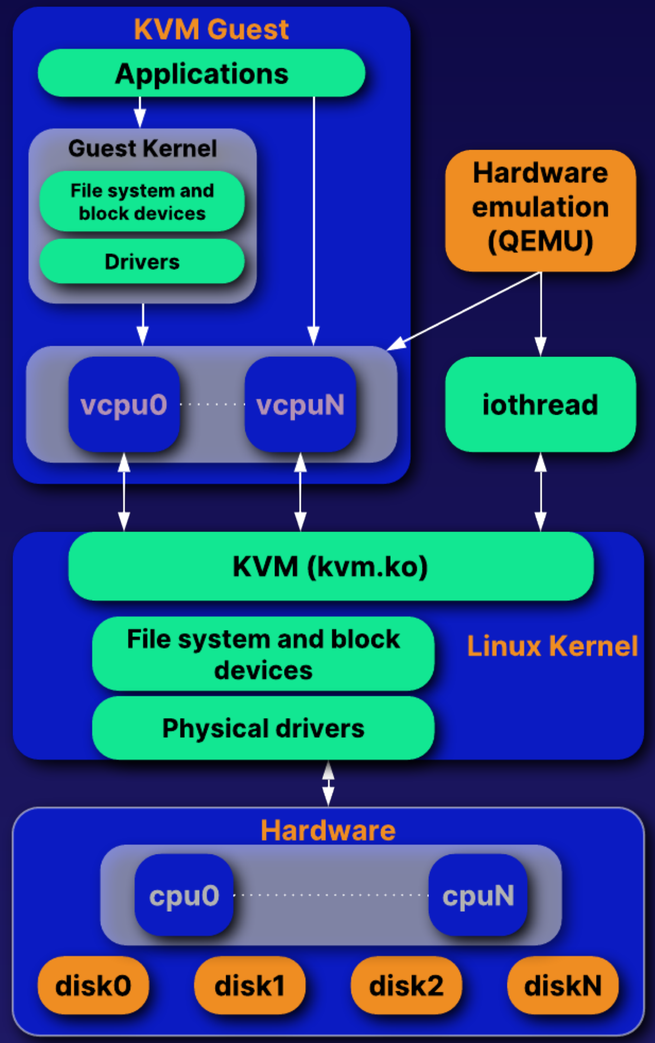


The hypervisor provides the resources that the guest operating system requires:

- Virtual CPU(s)
- Virtual Memory
- Virtual Disk(s)
- Virtual Networking

Other resources:
- Virtual media:
  - CD/DVD/ISO images
  - USB
  - Sound
- Video

Mapped physical resources:
- Networking:
  - IP
  - Storage
- GPU(s)



The hypervisor provides the resources that the guest operating system requires:

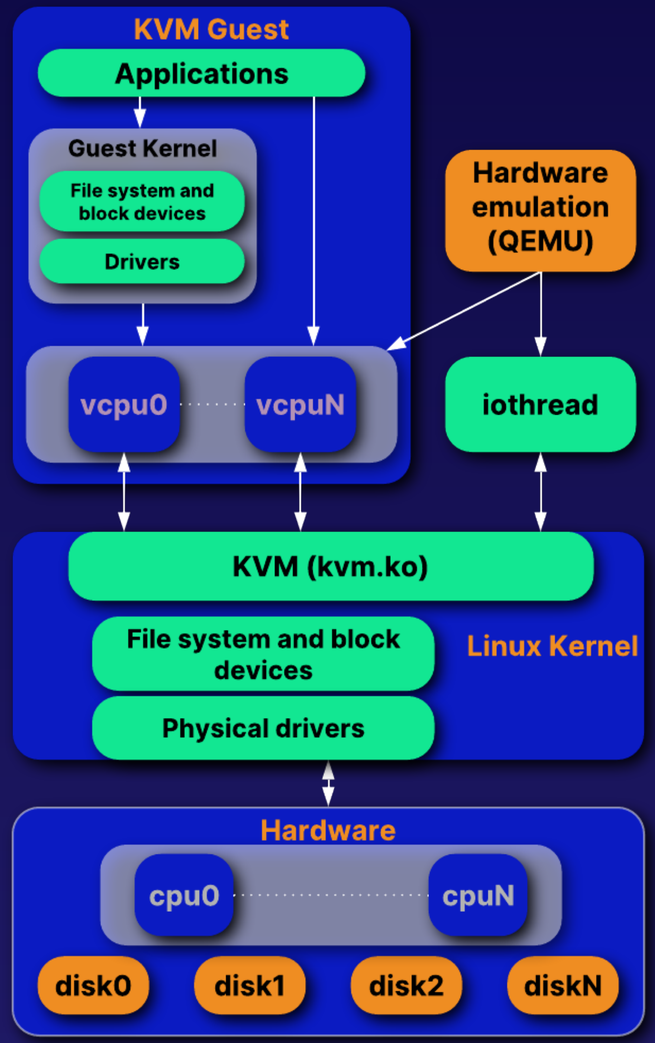
- Virtual CPU(s)
- Virtual Memory
- Virtual Disk(s)
- Virtual Networking

Other resources:

- Virtual media:
  - CD/DVD/ISO images
  - USB
  - Sound
- Video

Mapped physical resources:

- Networking:
  - IP
  - Storage
- GPU(s)



The hypervisor provides the resources that the guest operating system requires:

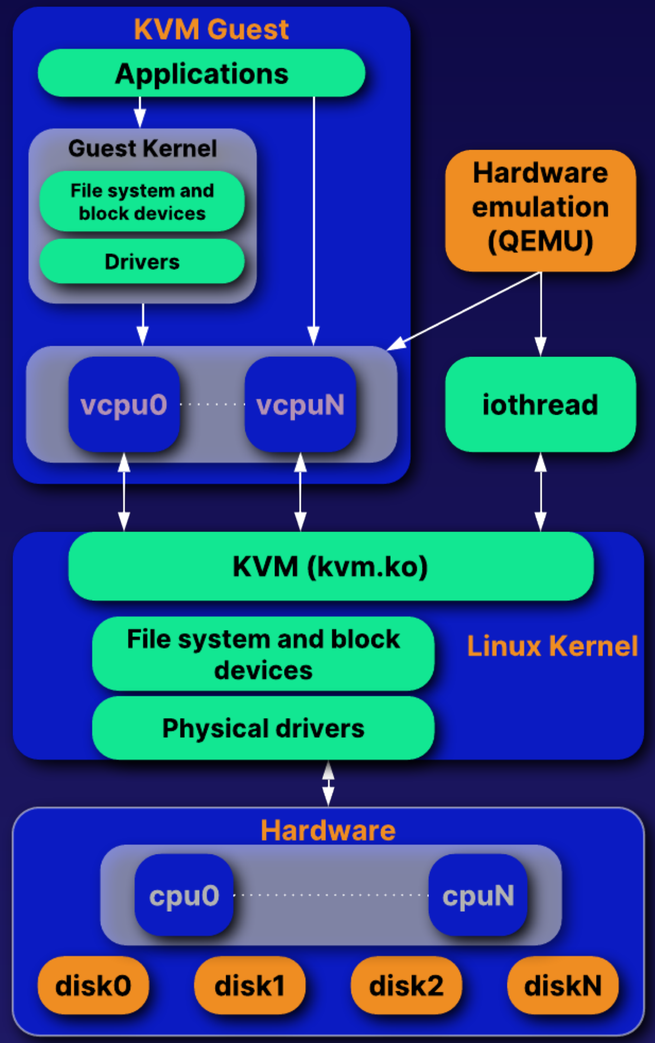
- Virtual CPU(s)
- Virtual Memory
- Virtual Disk(s)
- Virtual Networking

Mapped physical resources:

- Networking:
  - IP
  - Storage
- GPU(s)

Other resources:

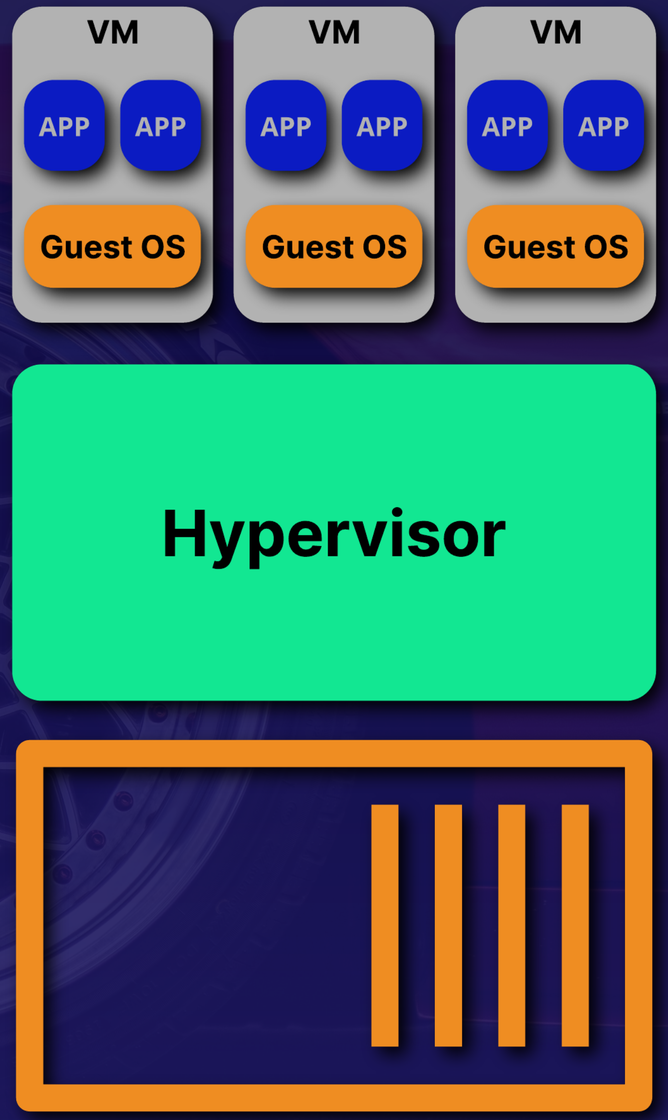
- **Virtual media:**
  - CD/DVD/ISO images
  - USB
  - Sound
- **Video**



Type-1 vs. Type-2 Hypervisors

Type-1 Hypervisor:

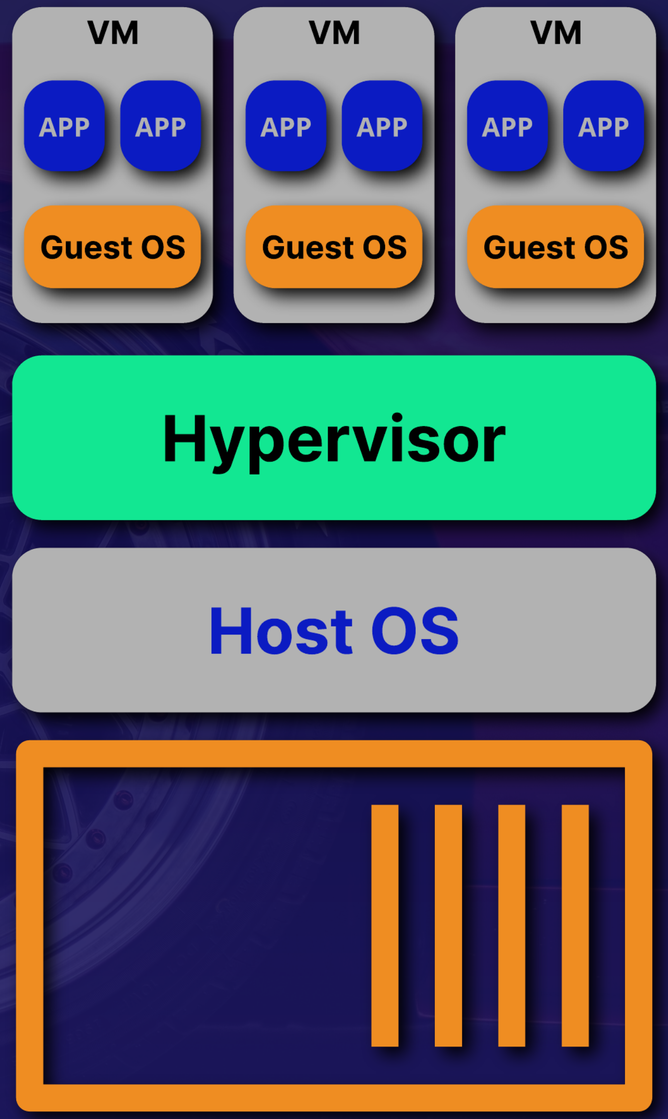
- Sits directly on the hardware
- It *is* the host **operating system**
- Example: VMWare ESX



Type-1 vs. Type-2 Hypervisors

Type-2 Hypervisor:

- Runs on top of a host operating system
- Example: VirtualBox



Types of Virtualization

Server/Operating System Virtualization

Run multiple operating systems on one or more physical server(s):

- More efficient hardware utilization:
  - Consolidate workloads for efficiency
- A virtual machine is independent of the hardware
- Virtual machines can be moved around for hardware maintenance, with minimal disruption
- Virtual machines can be replicated to remote data centers or the cloud:
  - Disaster recovery
  - Capacity needs
- Single means of managing servers
- Automation:
  - Deployment
  - Lifecycle management

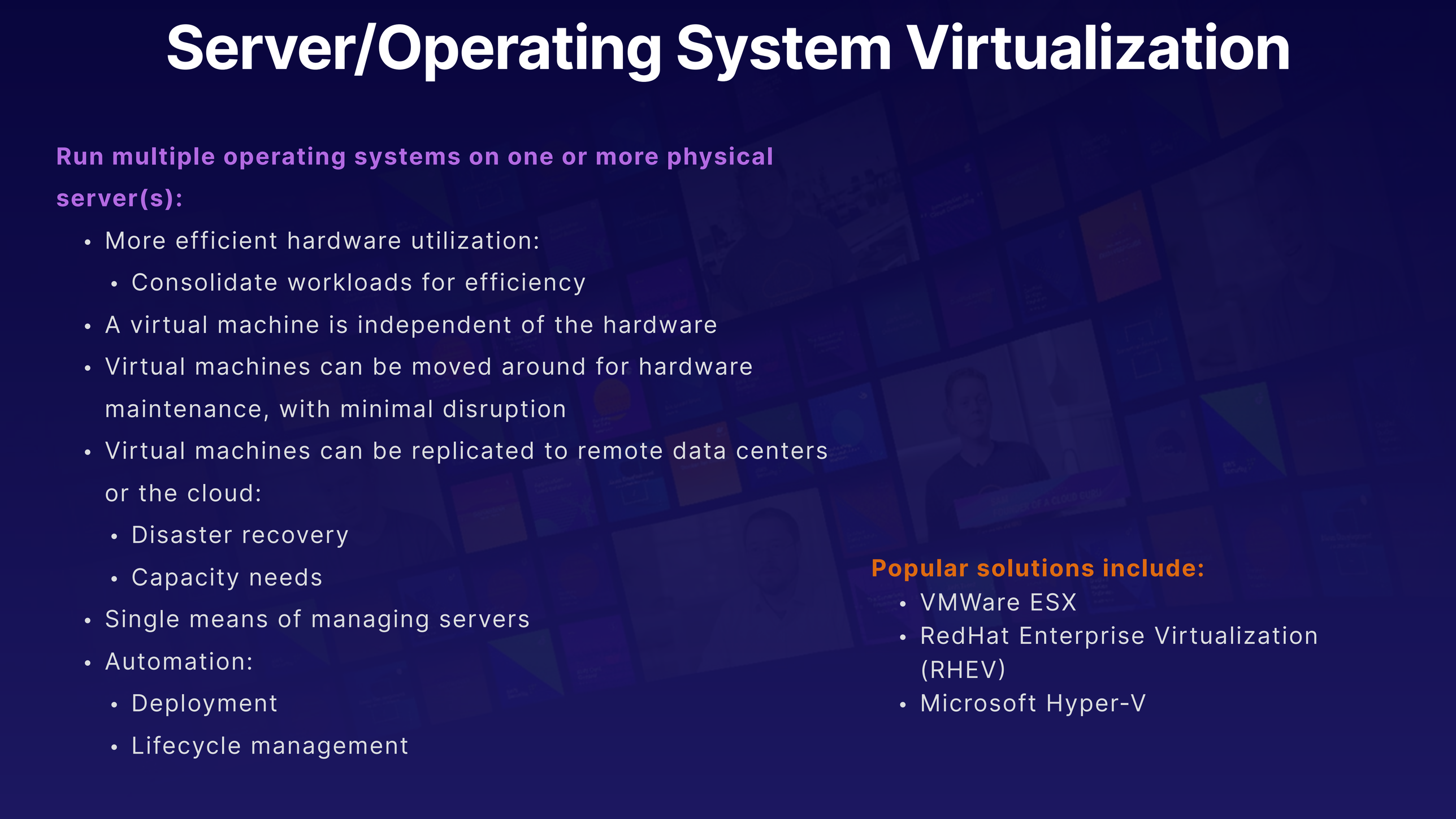
Popular solutions include:

- VMWare ESX
- RedHat enterprise virtualization (rhev)
- Microsoft Hyper-V

Server/Operating System Virtualization

Run multiple operating systems on one or more physical server(s):

- More efficient hardware utilization:
  - Consolidate workloads for efficiency
- A virtual machine is independent of the hardware
- Virtual machines can be moved around for hardware maintenance, with minimal disruption
- Virtual machines can be replicated to remote data centers or the cloud:
  - Disaster recovery
  - Capacity needs
- Single means of managing servers
- Automation:
  - Deployment
  - Lifecycle management



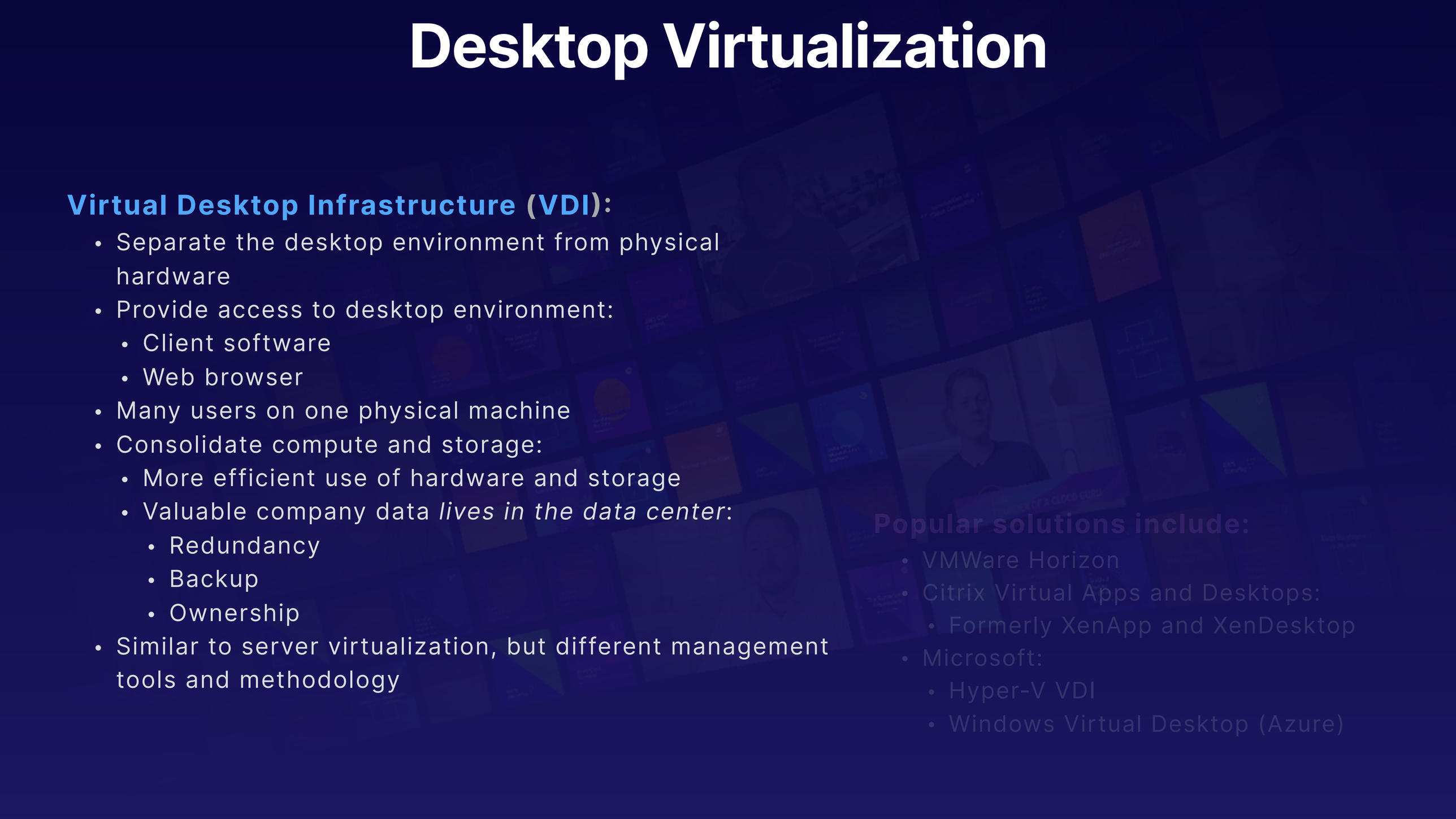
Popular solutions include:

- VMWare ESX
- RedHat Enterprise Virtualization (RHEV)
- Microsoft Hyper-V

Desktop Virtualization

Virtual Desktop Infrastructure (VDI):

- Separate the desktop environment from physical hardware
- Provide access to desktop environment:
  - Client software
  - Web browser
- Many users on one physical machine
- Consolidate compute and storage:
  - More efficient use of hardware and storage
  - Valuable company data *lives in the data center*:
    - Redundancy
    - Backup
    - Ownership
- Similar to server virtualization, but different management tools and methodology



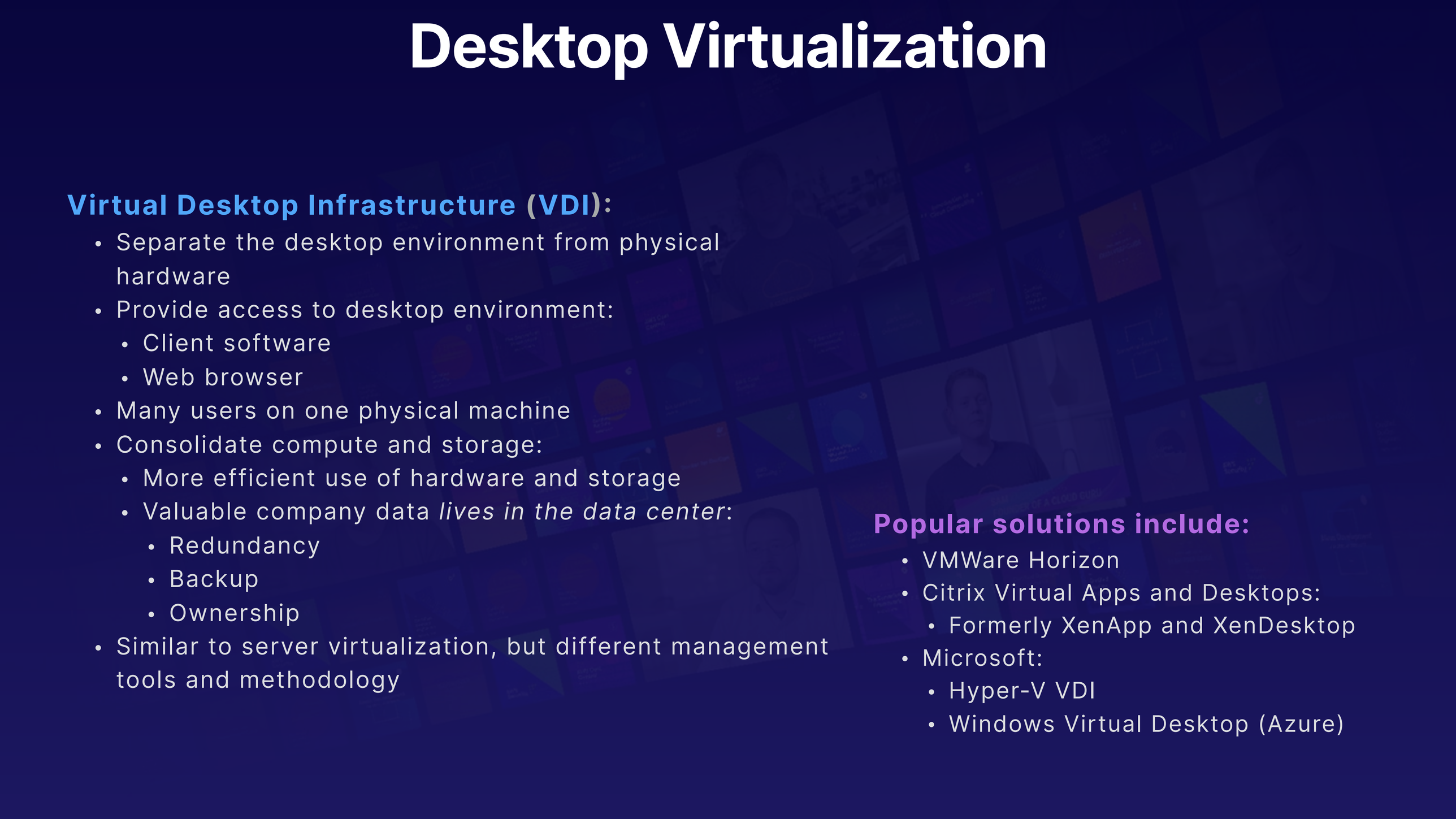
Popular solutions include:

- VMWare Horizon
- Citrix Virtual Apps and Desktops:
  - Formerly XenApp and XenDesktop
- Microsoft:
  - Hyper-V VDI
  - Windows Virtual Desktop (Azure)

Desktop Virtualization

Virtual Desktop Infrastructure (VDI):

- Separate the desktop environment from physical hardware
- Provide access to desktop environment:
  - Client software
  - Web browser
- Many users on one physical machine
- Consolidate compute and storage:
  - More efficient use of hardware and storage
  - Valuable company data *lives in the data center*:
    - Redundancy
    - Backup
    - Ownership
- Similar to server virtualization, but different management tools and methodology



Popular solutions include:

- VMWare Horizon
- Citrix Virtual Apps and Desktops:
  - Formerly XenApp and XenDesktop
- Microsoft:
  - Hyper-V VDI
  - Windows Virtual Desktop (Azure)

Application Virtualization

Containers

- Lightweight version of virtualization:
  - Multiple isolated user-space instances
- Only provide the *bare minimum* needed:
  - Not a full OS install
- Often looks like “real,” full operating systems
- Certain parts of the host operating system are shared

- **Examples:**
  - **Docker**
  - **LXC**
  - **Solaris Containers/Zones**
  - **FreeBSD jail(s)**
  - **chroot jail(s)**

Application Virtualization

Containers

- Lightweight version of virtualization:
  - Multiple isolated user-space instances
- Only provide the *bare minimum* needed:
  - Not a full OS install
- Often looks like “real,” full operating systems
- Certain parts of the host operating system are shared

- **Examples:**
  - **Docker**
  - **LXC**
  - **Solaris Containers/Zones**
  - **FreeBSD jail(s)**
  - **chroot jail(s)**

Network (Function) Virtualization

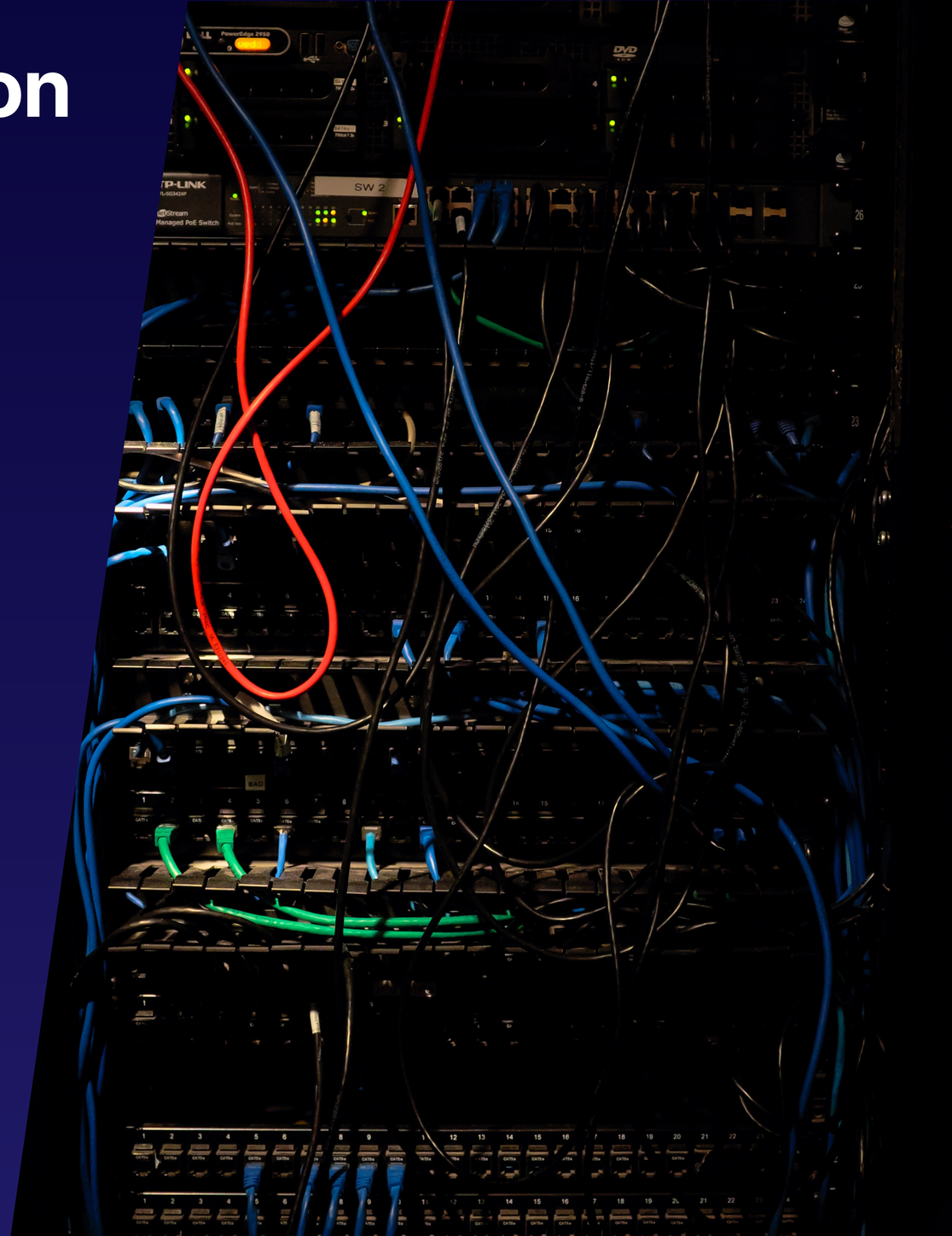
Virtual representation of physical networking components:

- Logical ports
- Switches
- Routers
- VPN(s)
- Firewalls
- Load Balancers(s)

Can be deployed alongside virtual servers:

- Deploy an entire unit of virtual infrastructure at the same time:
  - Automation
  - Scaling

FAST AND EASY TO DEPLOY



Network (Function) Virtualization

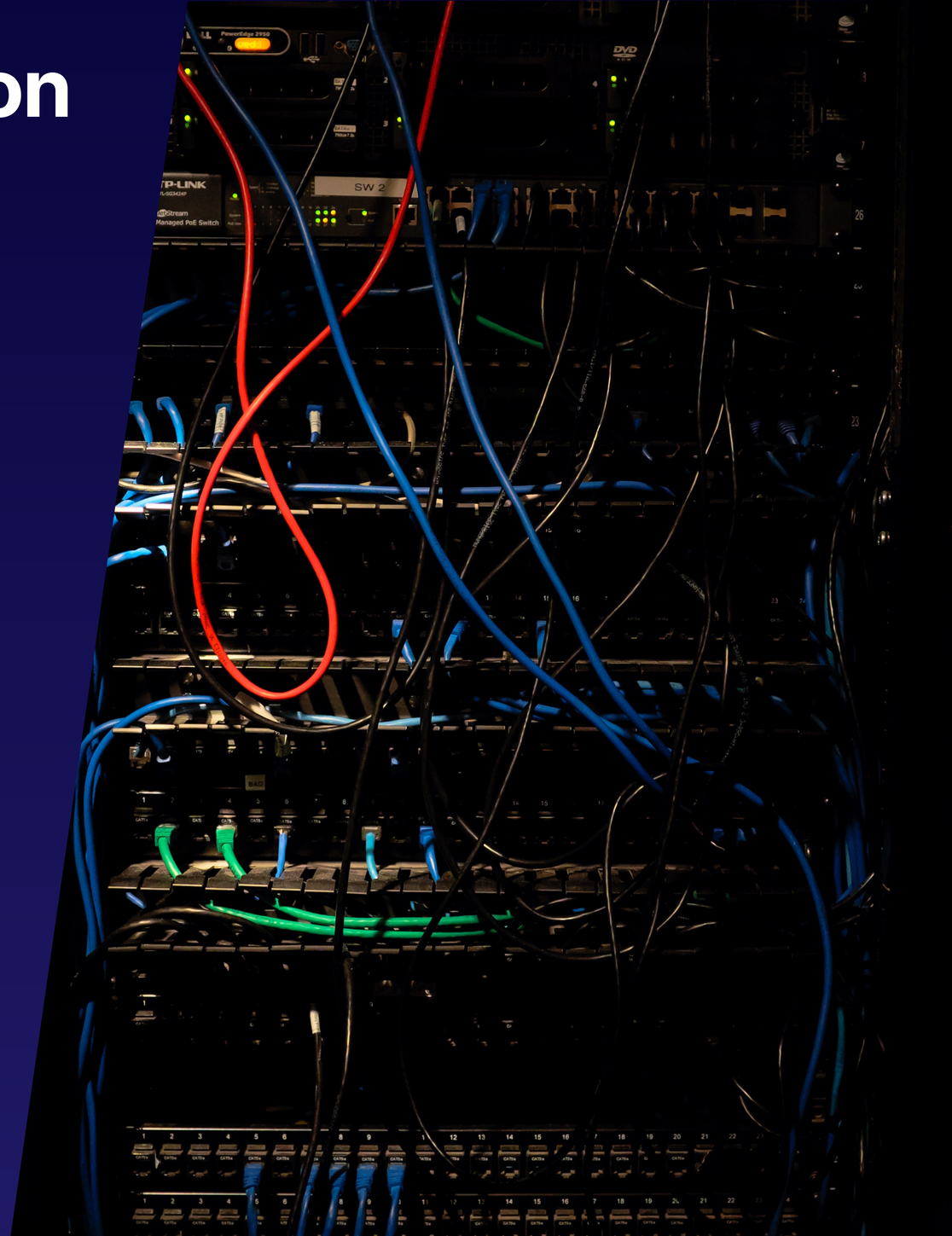
Virtual representation of physical networking components:

- Logical ports
- Switches
- Routers
- VPN(s)
- Firewalls
- Load Balancers(s)

Can be deployed alongside virtual servers:

- Deploy an entire unit of virtual infrastructure at the same time:
  - Automation
  - Scaling

FAST AND EASY TO DEPLOY



Network (Function) Virtualization

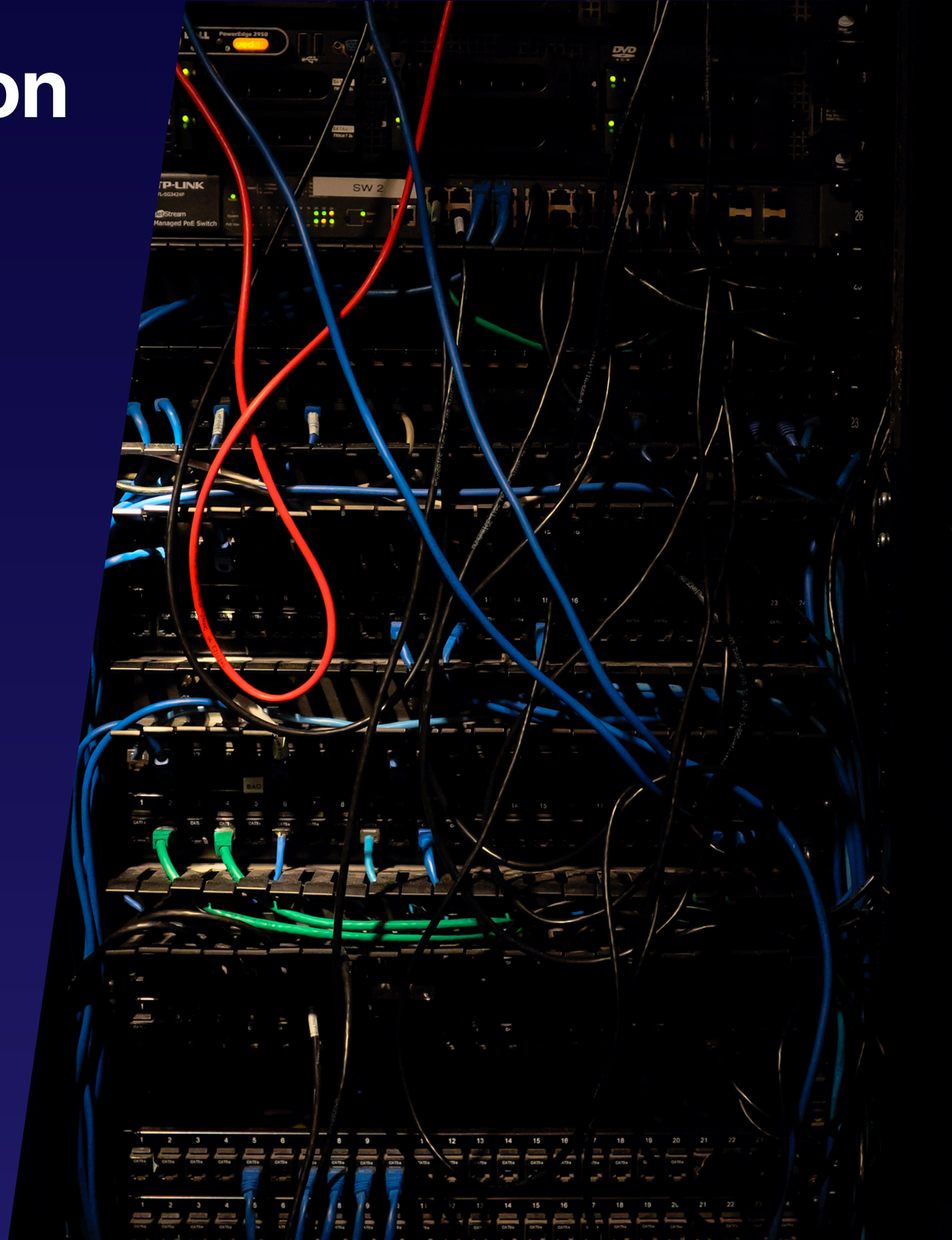
Virtual representation of physical networking components:

- Logical ports
- Switches
- Routers
- VPN(s)
- Firewalls
- Load Balancers(s)

Can be deployed alongside virtual servers:

- Deploy an entire unit of virtual infrastructure at the same time:
  - Automation
  - Scaling

FAST AND EASY TO DEPLOY



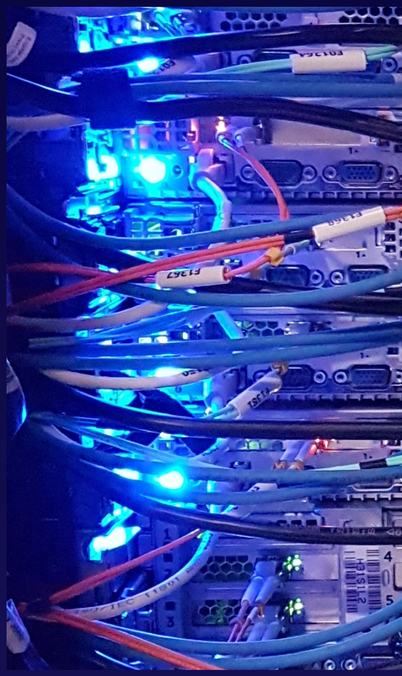
In this lesson, we covered:

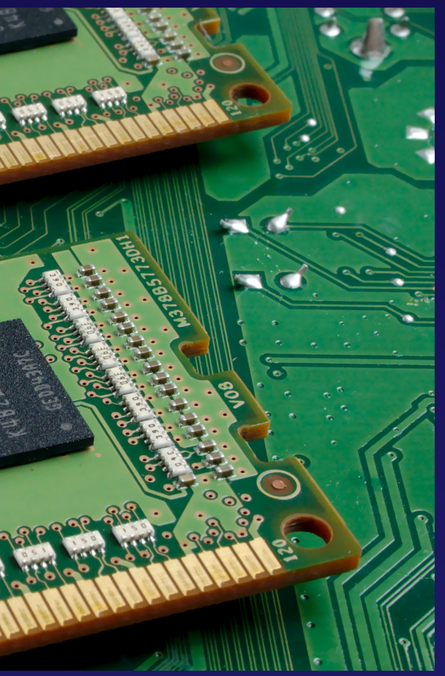


- **What** virtualization is
- A brief **history** of virtualization
- **How** virtualization works
- What a **hypervisor** is:
  - **Type-1** vs. **Type-2** hypervisors
- Different **types** of virtualization

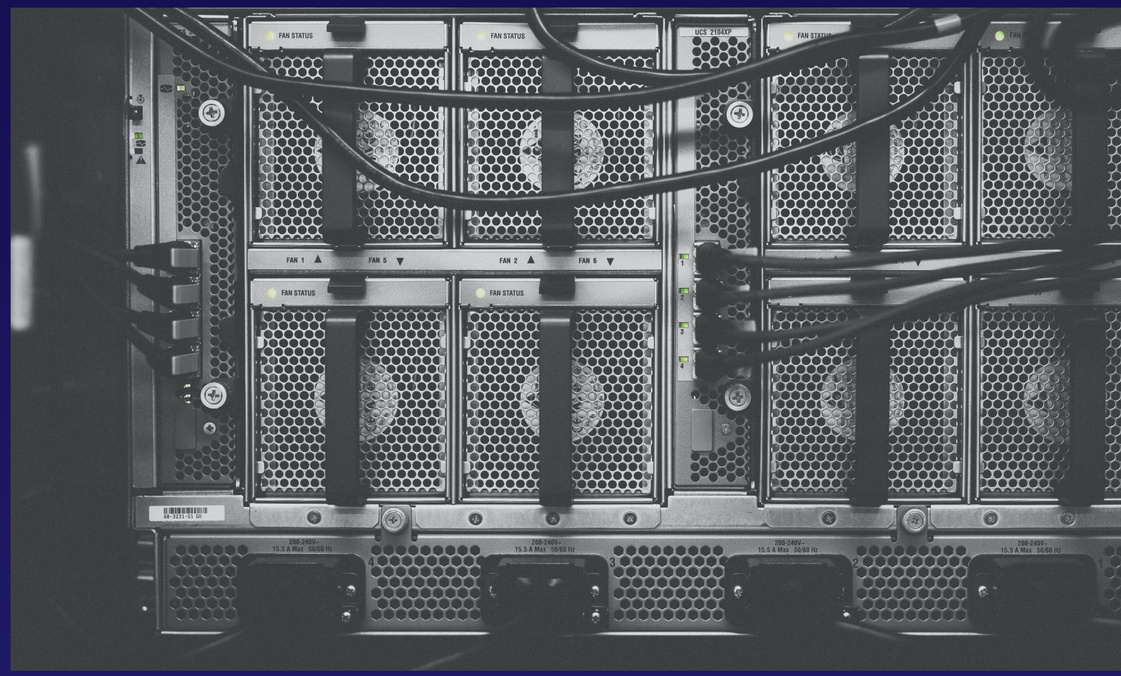
Tom Dean
Linux Training Architect

Why Virtualize?





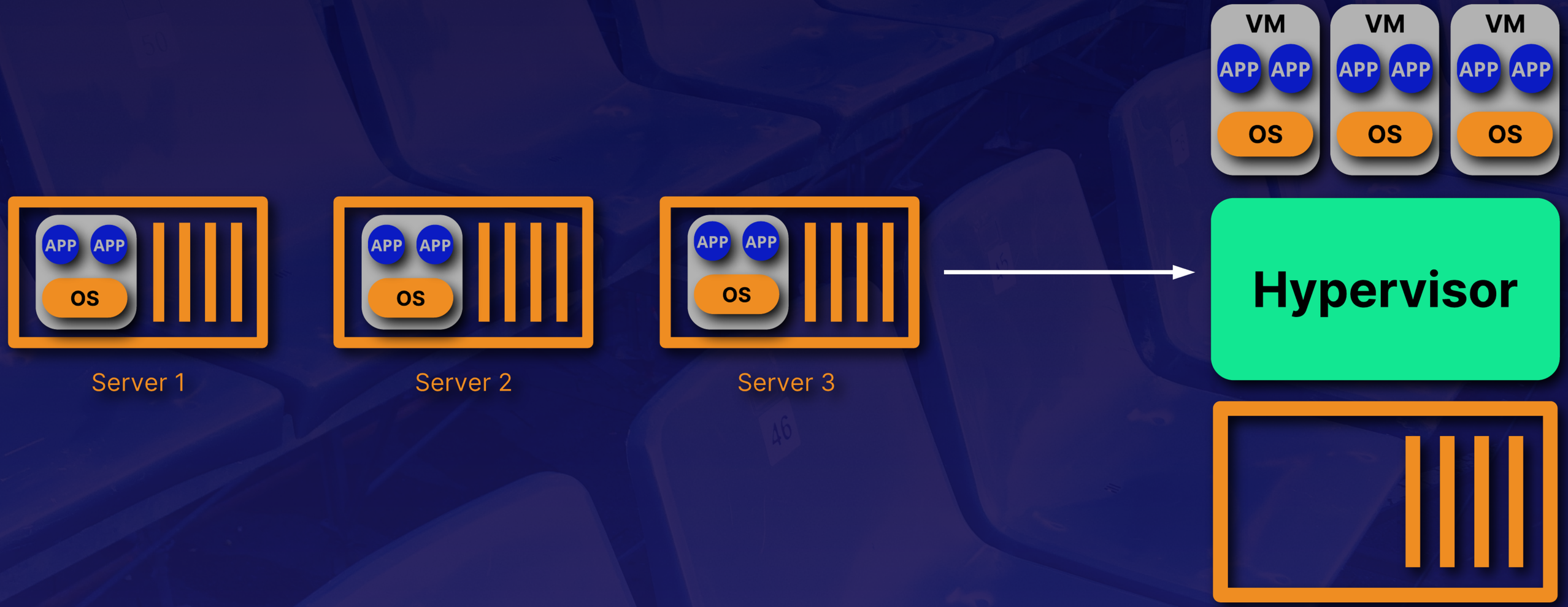
There are many reasons to consider virtualizing some or all of your physical infrastructure. Here are some of the more prominent ones.



Reduce Your Hardware Costs

Generally, most traditional *physically deployed* server footprints average 5-15% utilization. **It's inefficient.**

Virtualization allows you to decouple your services and applications from the physical (server) layer, and deploy it more efficiently and reliably than the traditional 1:1 service/application-to-physical server relationship.



Virtualization creates a **virtual machine** for the guest operating system to live within, with *virtual resources* instead of *physical resources*. Physical resources can also be mapped to a guest where necessary.

By virtualizing, you can:

✓ Retire older hardware

- Retire older servers, which are more expensive to operate and cost more to support
  - *Less equipment* for staff to maintain

✓ Standardize

- Standardize on a single (or minimal number of) hardware configuration(s)
  - Keep spares on-site for quick repair turnaround
  - Reduces complexity of the hardware layer design
  - Reduces maintenance complexity
  - Use commodity, "bang for the buck" hardware

By virtualizing, you can:

✓ **Retire older hardware**

- Retire older servers, which are more expensive to operate and cost more to support
  - *Less equipment* for staff to maintain

✓ **Standardize**

- Standardize on a single (or minimal number of) hardware configuration(s)
  - Keep spares on-site for quick repair turnaround
  - Reduces complexity of the hardware layer design
  - Reduces maintenance complexity
  - Use commodity, “bang for the buck” hardware

Reduce Your Energy Costs

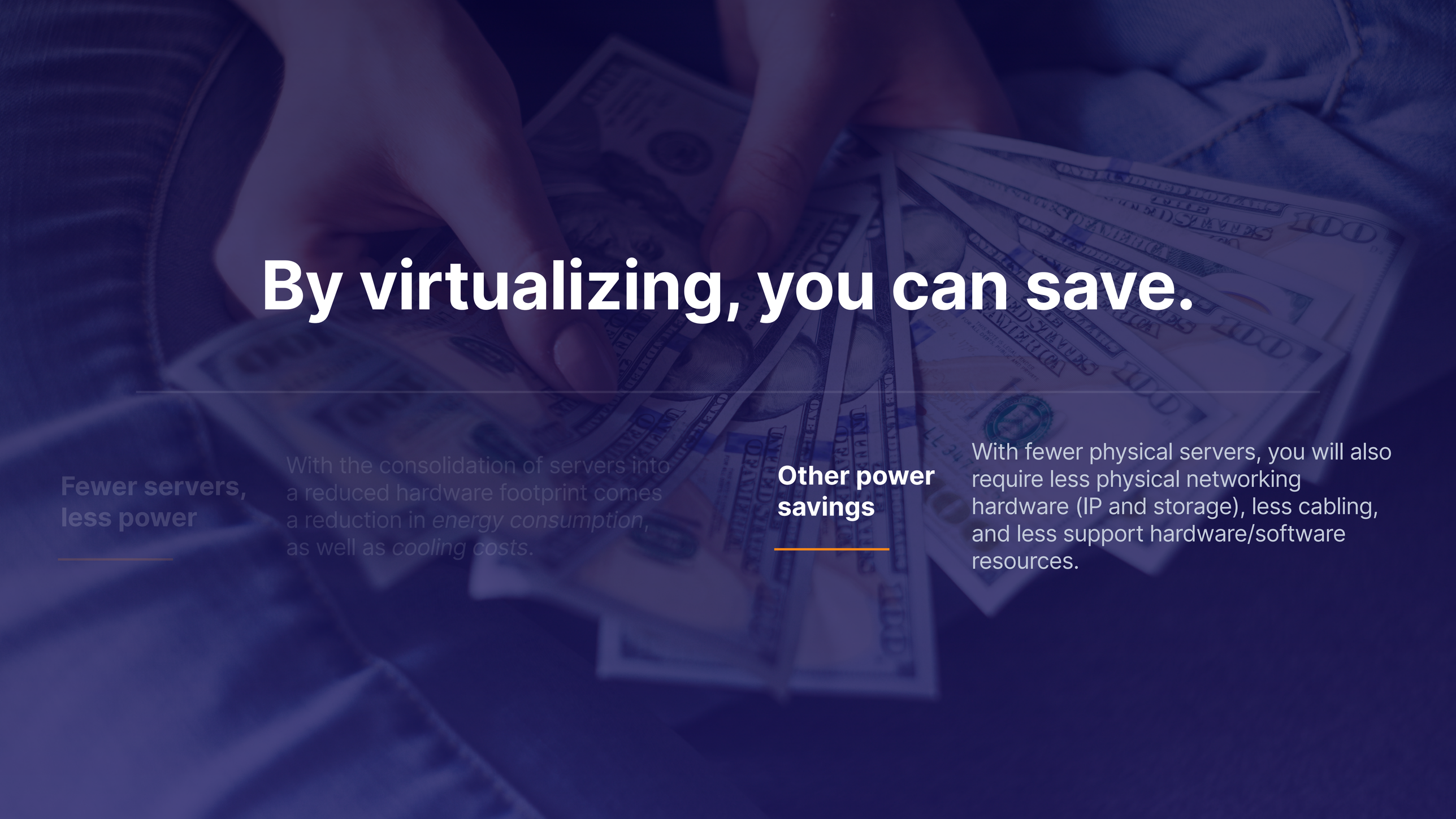
By virtualizing, you can save.

Fewer servers, less power

With the consolidation of servers into a reduced hardware footprint comes a reduction in *energy consumption*, as well as *cooling costs*.

Other power savings

With fewer physical servers, you will also require less physical networking hardware (IP and storage), less cabling, and less support hardware/software resources.



By virtualizing, you can save.

Fewer servers, less power

With the consolidation of servers into a reduced hardware footprint comes a reduction in *energy consumption*, as well as *cooling costs*.

Other power savings

With fewer physical servers, you will also require less physical networking hardware (IP and storage), less cabling, and less support hardware/software resources.

Reduce Your Footprint

Fewer physical servers means less rack space.

Co-Located Facility

If you're in a hosted facility and are paying by the rack, this can mean big savings.

Own Facility

If you own your datacenter, you can use the floor space for other things.

Fewer physical servers means less rack space.

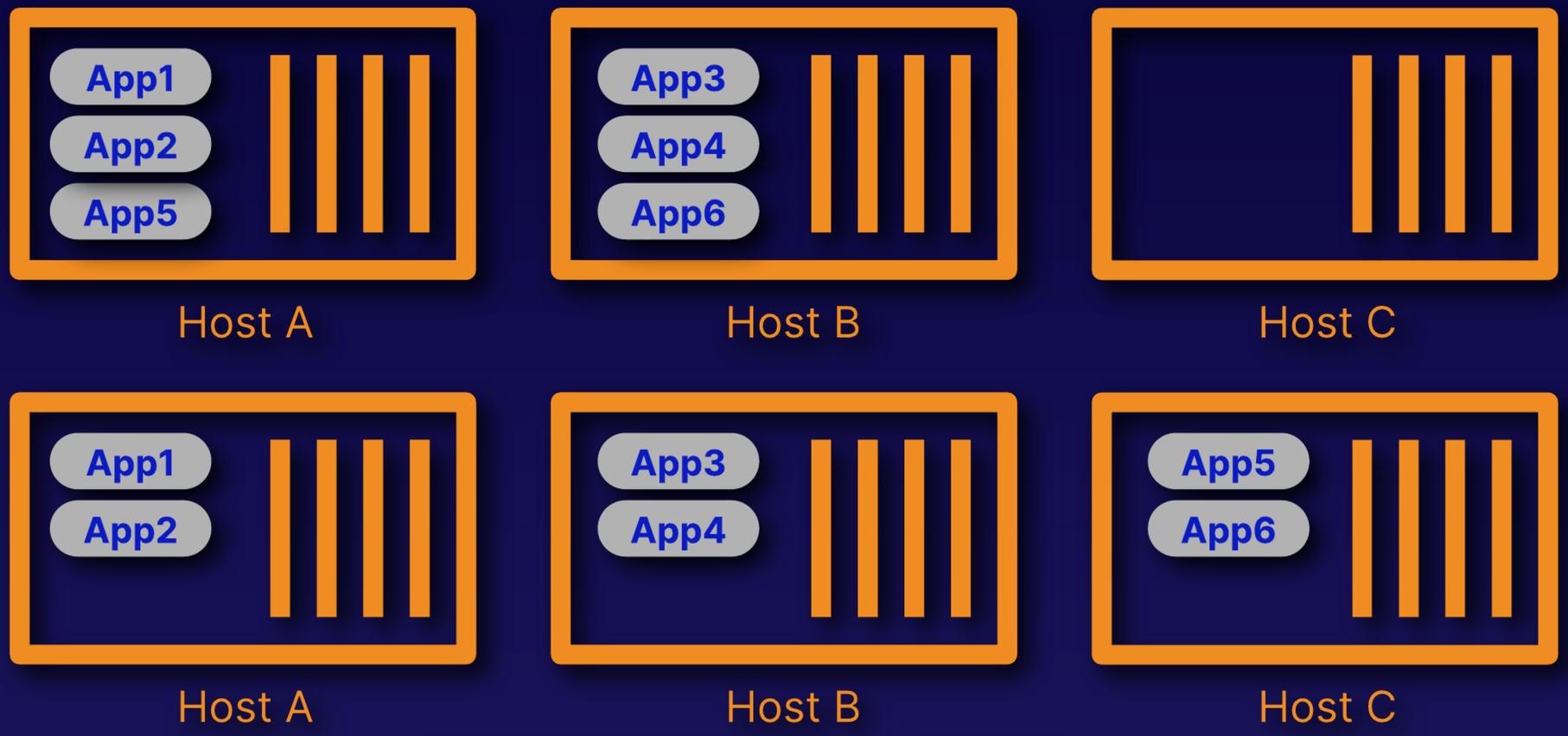
Co-Located Facility

If you're in a hosted facility and are paying by the rack, this can mean big savings.

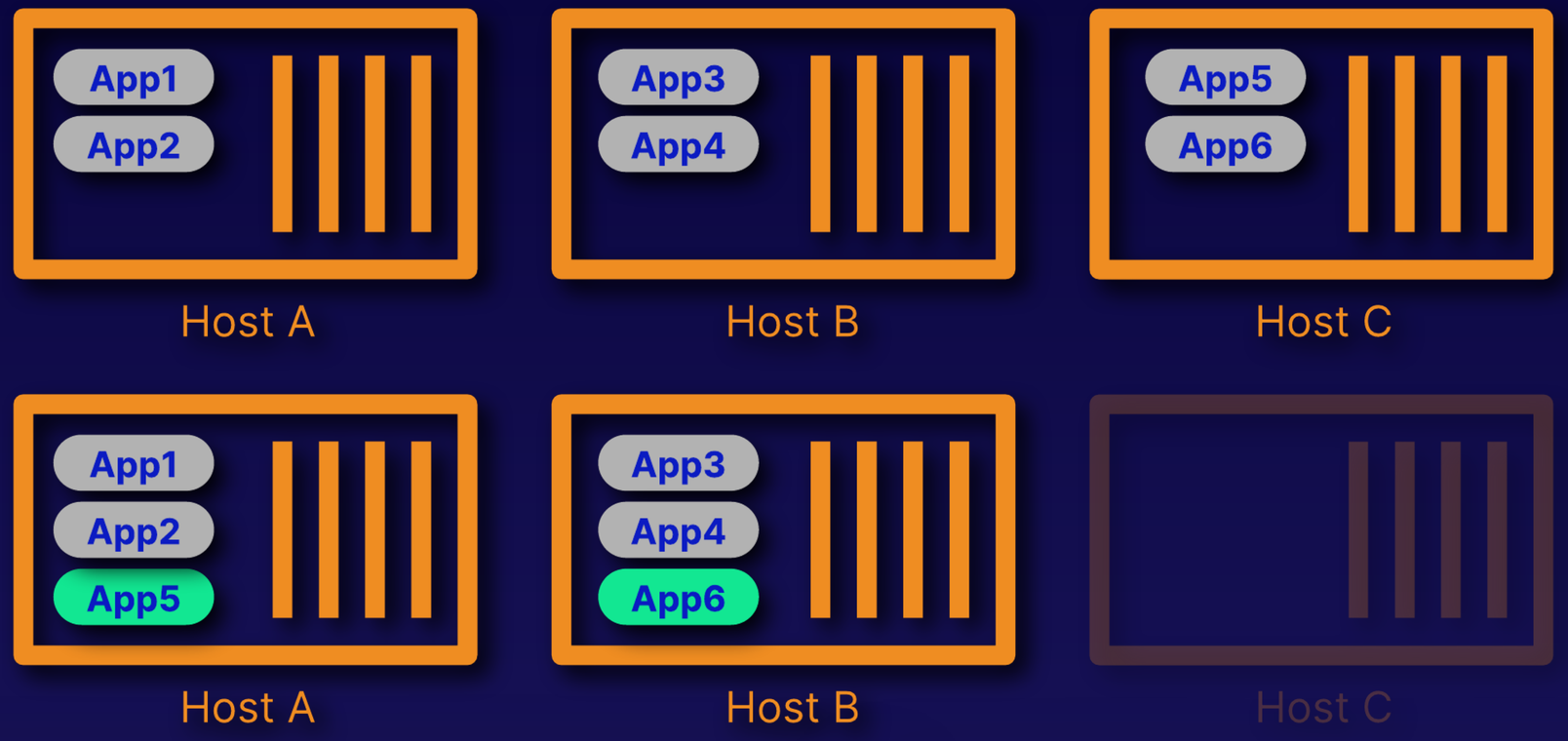
Own Facility

If you own your data center, you can use the floor space for other things.

Workload Management/ Fault Tolerance/ Maintenance



In an **enterprise-level virtualization solution**, you can move virtual machines between hosts with no (or minimal) interruption. This facilitates moving guests around to **balance workloads** across hosts.



In a **properly designed virtual infrastructure**, there will be enough virtualization hosts so that one or more can be taken offline for maintenance while all the guests can continue to run on the remaining hypervisor(s).

In addition, most enterprise-level virtualization solutions, using shared storage, will support *moving guests to another hypervisor* in the event of an **issue or outage**.