NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

# Servers & Virtualization

Cloud Infrastructure Engineering

**Nanyang Technological University**
**& Skills Union - 2022/2023**

# Course Content

- Quick Check-In
- Dive into the basics of Servers
- Dive into the basics of Virtualization
- Explore the uses cases for different Server types
- Differentiate between Virtualization & Containerization

| Time | What | How or Why |
|---|---|---|
| 7:15pm - 7:30pm | Part 1 - Presentation | General Information on Servers |
| 7:30pm - 7:55pm | Part 2 - Presentation | Types of Servers |
| 7:55pm - 8:05pm | Break | |
| 8:05pm - 8:20pm | Part 3 - Presentation | General Information on Virtualization |
| 8:20pm - 8:40pm | Part 4 - Presentation | Virtualization vs Containerization |
| 8:40pm - 8:45pm | Assignment Briefing | |
| 8:45pm - 9:45pm | Learners attempt assignments | |
| 9:45pm - 10:00pm | Wrap Up | |

# Recap

- Memory
  - RAM
- Storage
  - Magnetic, Optical, Flash, Paper & Cloud
- RAID
  - RAID 0 - FASTEST, blocks striped
  - RAID 1 - SAFEST, blocks mirrored
  - RAID 4 - FAST & SAFE, blocks striped & dedicated parity
  - RAID 5 - FAST & SAFE, blocks striped & distributed parity
  - RAID 6 - FAST & SAFE, blocks striped & 2 distributed parity
  - RAID 10 - FASTEST & SAFEST, blocks mirrored & striped

# Self Study Check-In

# Q1) What are some types of servers that you are aware of?

Q2) What are some server virtualization types that you know of?

Q3) What is the purpose of virtualization in the context of server infrastructure?

# Servers

# What is a Server?

The term ‘server’ has **double meaning** in IT.

It is used to describe a **computer** that **makes resources available over a network**, as well as the **program that runs on this computer (Web Servers)**.

So it’s necessary to have two different server definitions:

# What is a Server?

Server (Hardware):

a hardware-based server is a **physical device connected to a computer network** on which one or more software-based servers run alongside the operating system.

An alternative term for a hardware-based server is **host**. In principle, any computer can be used as a host with server software.

# What is a Server?

Server (Software):

a software-based server is a **program that offers a specialized service to be used by other programs** (known as **clients**) locally or via a network.

The service offered depends on the type of server software. The client-server model is the basis of communication. When it comes to exchanging data, service-specific communication protocols are used.

# How Do Servers Work?

**Server services operating via computer networks** are realized with the help of **client-server models**.

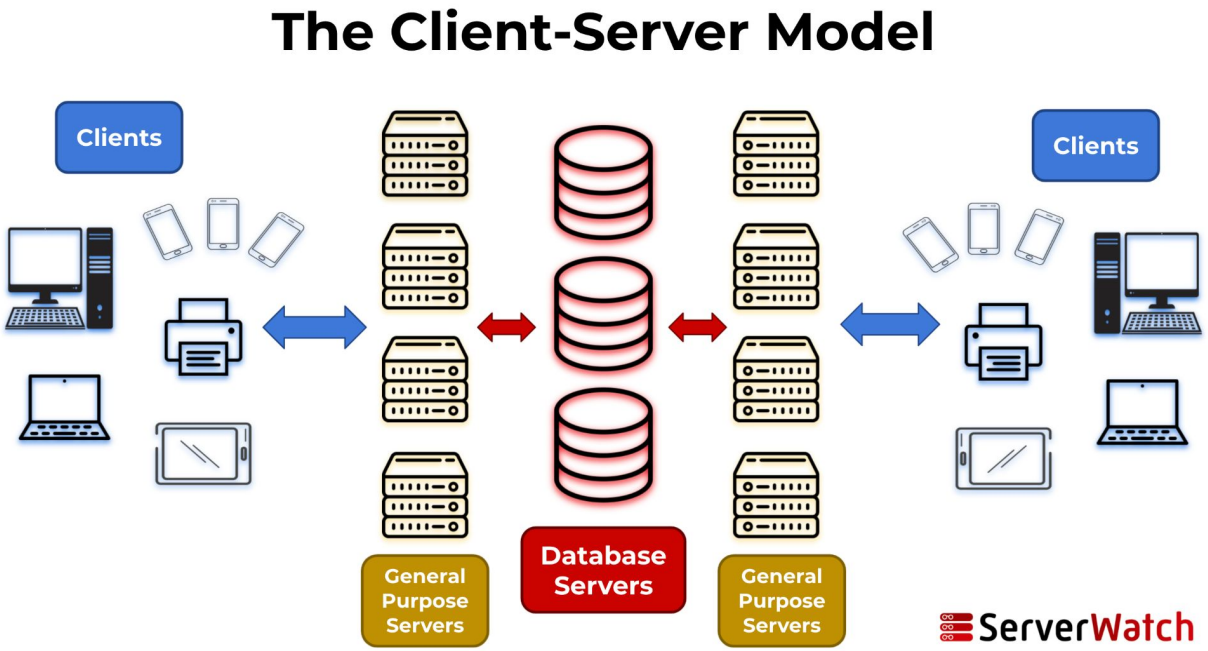
This concept makes it possible to **divide tasks among different computers** and allow users access to them simultaneously.

# How Do Servers Work?

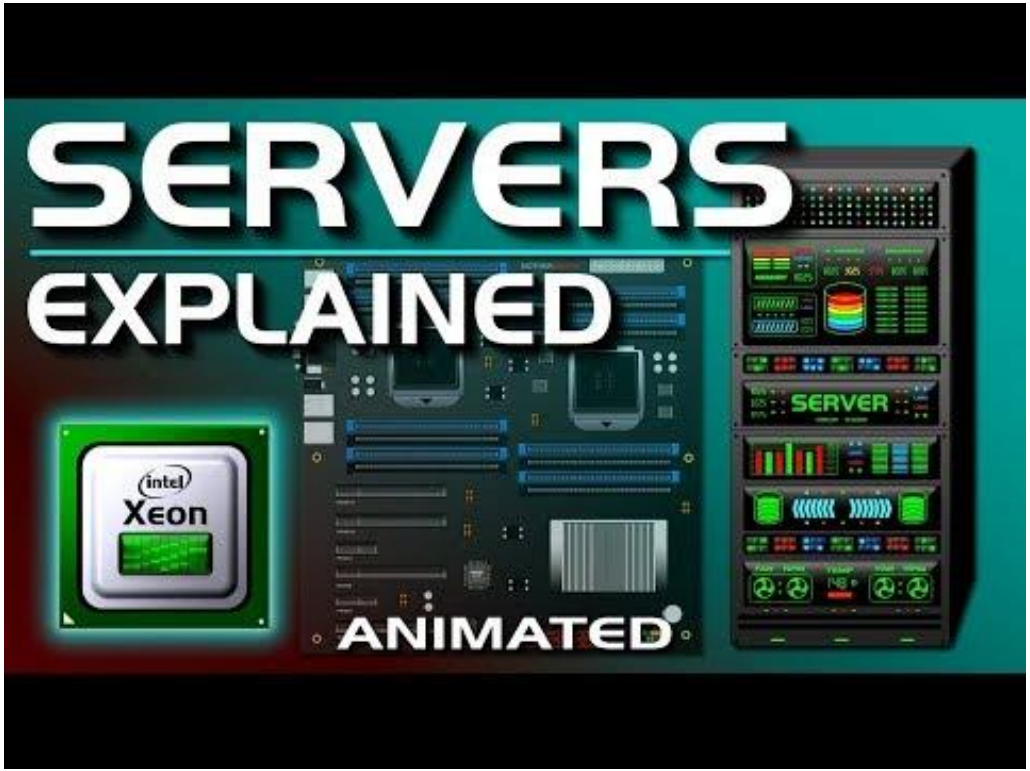
Every service provided via a network **requires a server (software)**, which is permanently **on standby**.

This is the only way to ensure that clients, like web browsers or e-mail programs, always have the opportunity to **access the server and utilize its service when needed**.

# How Do Servers Work?



# How Do Servers Work? - Video



# Types of Servers

# What Types of Servers Are There?

## By Server Functionality

Different types of servers **play one or multiple jobs**, from serving email and multimedia content to protecting internal networks and hosting web applications.

The most common types of functionality include servers for shared network files, databases, proxy security, email, and server backups.

# Server Functionality Types

- File Servers
- Print Servers
- Application Servers
- Web Servers
- DNS Servers
- Database Servers
- Proxy Servers
- Monitoring Servers
- Mail Servers

# File Servers

File servers **store and distribute files.**

Multiple clients or users may share files stored on a server.

In addition, centrally storing files **offers easier backup or fault tolerance** solutions than attempting to provide security and integrity for files on every device in an organization.

File server hardware can be designed to **maximize read and write speeds to improve performance**.

# File Servers


Amazon FSx

Azure File


Google Cloud Storage

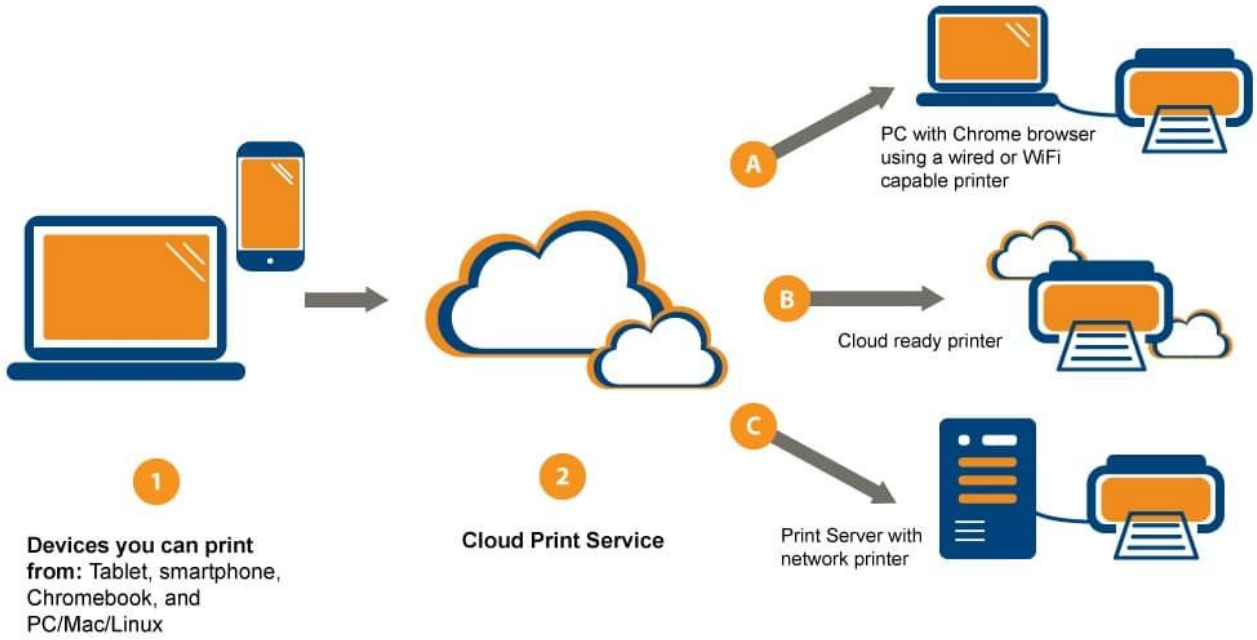
# Print Servers

Print servers allow for the **management and distribution** of **printing functionality**.

Rather than attaching a printer to every workstation, a **single print server can respond to printing requests from numerous clients**.

Today, some larger and higher-end printers come with their own built-in print server, which removes the need for an additional computer-based print server. This internal print server also functions by responding to print requests from a client.

# Print Servers



# Print Servers

# Application Servers

Run applications in-lieu of client computers running applications locally.

Application servers often **run resource-intensive applications that are shared by a large number of users**.

Doing so removes the need for each client to have sufficient resources to run the applications. It also removes the need to install and maintain software on many machines as opposed to only one.

# Application Servers

# Web Servers

A web server is a special kind of application server that **hosts programs and data requested by users across the Internet or an intranet**.

Web servers **respond to requests from browsers running on client** computers for web pages, or other web-based services.

Common web servers include **Apache web servers**, Microsoft Internet Information Services (IIS) servers and **Nginx servers**.

# Web Servers

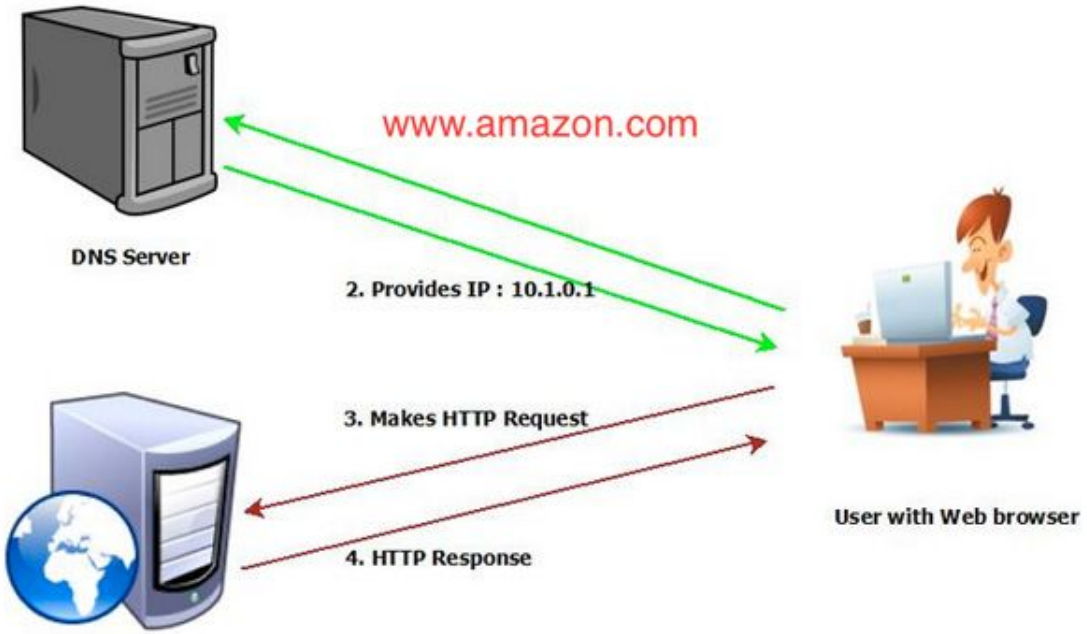


# DNS Servers

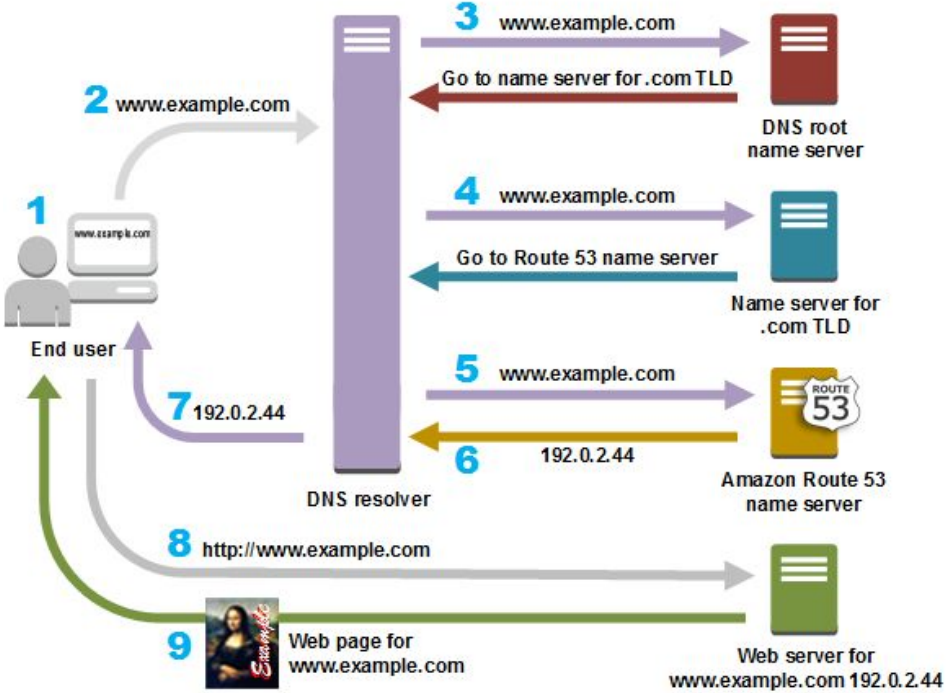
Domain Name System (DNS) servers are **application servers that provide name resolution to client computers by converting names easily understood by humans into machine-readable IP addresses**.

The DNS system is a widely distributed database of names and other DNS servers, each of which can be used to request an otherwise unknown computer name.

# DNS Servers


www.amazon.com
DNS Server
2. Provides IP : 10.1.0.1
3. Makes HTTP Request
4. HTTP Response
User with Web browser

# DNS Servers


1
End user
2 www.example.com
DNS resolver
3 www.example.com
Go to name server for .com TLD
DNS root name server
4 www.example.com
Go to Route 53 name server
Name server for .com TLD
5 www.example.com
6 192.0.2.44
Amazon Route 53 name server
7 192.0.2.44
8 http://www.example.com
9 Web page for www.example.com
Web server for www.example.com 192.0.2.44

# DNS Servers

# Database Servers

Databases need to be **accessible to multiple clients** at any given time and can require **extraordinary amounts of disk space**.

Both of these needs lend themselves well to locating such databases on servers. Database servers **run database applications** and respond to **numerous requests** from clients.

Common database server applications include Oracle, Microsoft SQL Server, DB2, and Informix.

# Database Servers

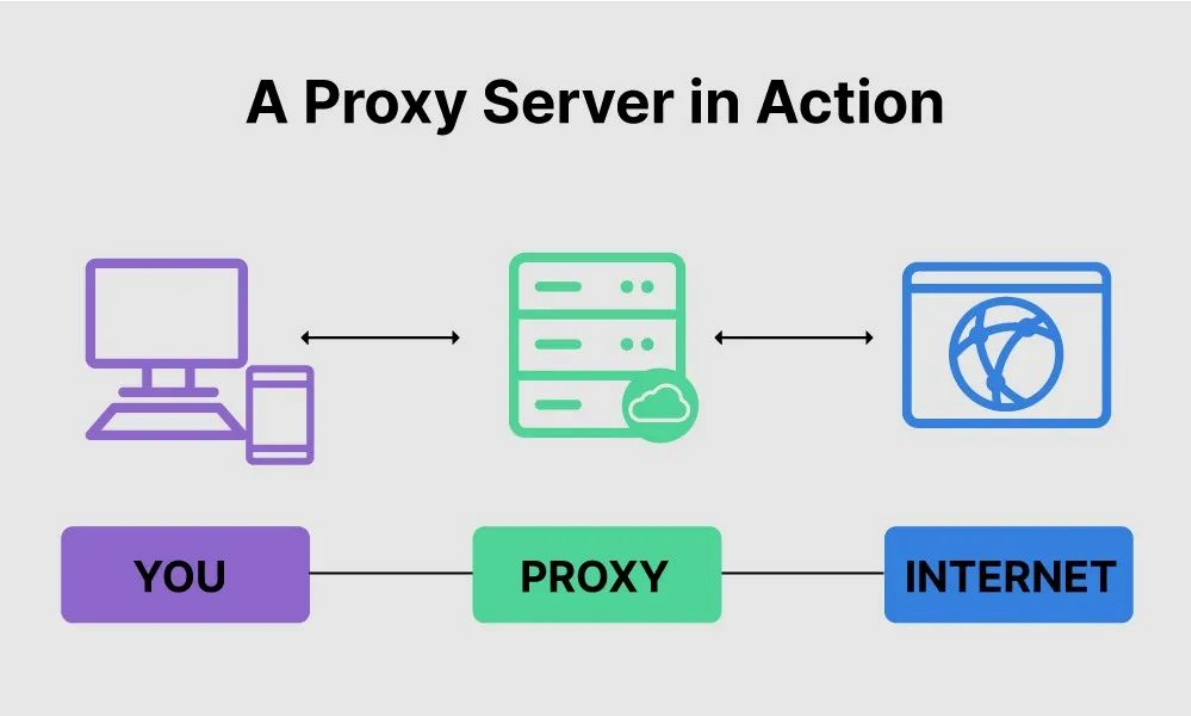



# Proxy Servers

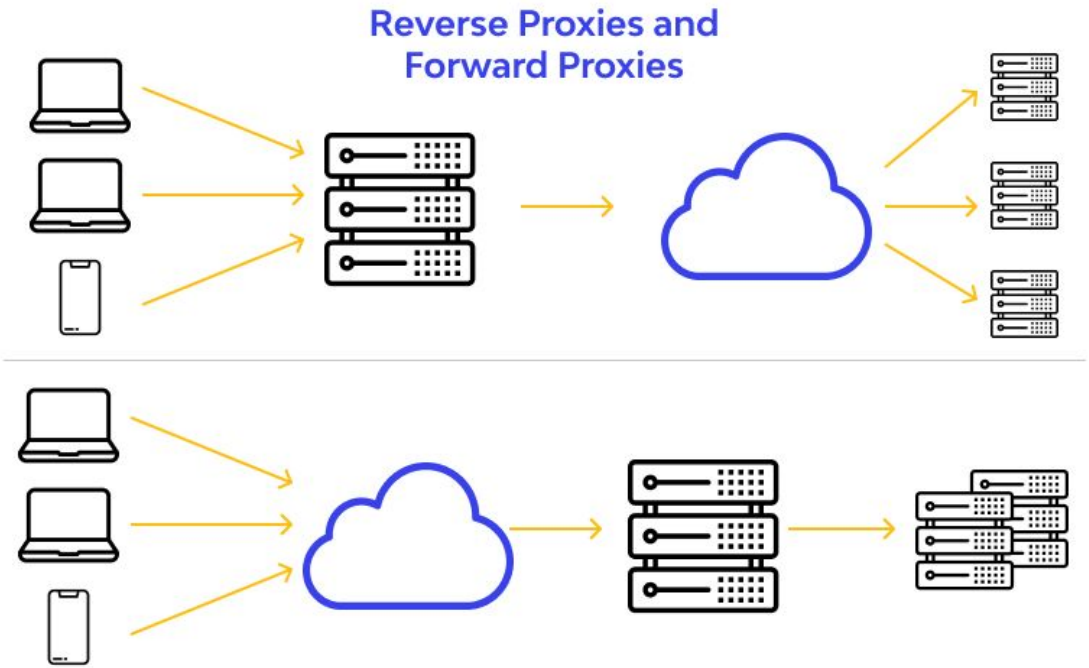
A proxy server acts as an **intermediary between a client and a server**.

Often used to isolate either the clients or servers for **security purposes**, a proxy server takes the request from the client.

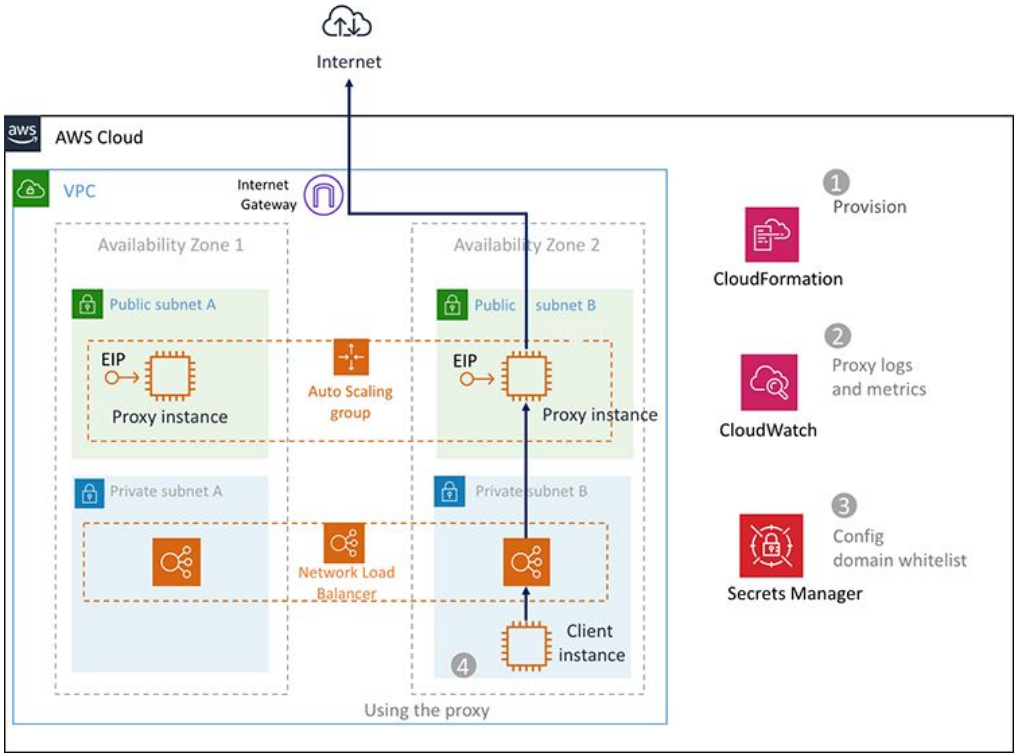
# Proxy Servers



# Proxy Servers



# Proxy Servers



# Monitoring Servers

**Monitoring server keep track of all the traffic on the network**, as well as the requests and replies of clients and servers, without interfering with those operations.

A monitoring server will **respond to requests from monitoring clients** such as those run by network administrators watching the health of the network.

# Monitoring Servers (In Practice)

In practice, most organizations **outsource** monitoring services to third-party tools.

This is done by **installing a lightweight agent** on your application, web and other **servers** to **retrieve logs and metrics to a centralized dashboard**

# Monitoring Servers

# Monitoring Servers






# Mail Servers

Mail servers **receive emails sent to a user and store them until requested by a client** on behalf of said user.

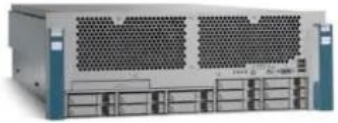
Having an email server allows for a **single machine to be properly configured and attached to the network at all times**. It is then ready to send and receive messages rather than requiring every client machine to have its own email subsystem continuously running.

# What Types of Servers Are There?

## By Form Factor

1. Tower Compute System
    a. **Vertical, standalone enclosure and least dense** with low maintenance and ideal for smaller organizations and teams.
2. Rack-Mounted Compute System
    a. **Mountable and less dense** but ideal for SMB to enterprise organizations that can **manage ongoing maintenance.**
3. Blade Compute System
    a. Offer the **densest build with a circuit board enclosure** that requires minimal cabling and maintenance.

# Form Factor Summary

| Tower Compute System | Rack-mounted Compute System | Blade Compute System |
| --- | --- | --- |
|  |  <br>  |  <br>  |

# Break Time

# Server Virtualization

# Server Virtualization

Process of **using software to divide physical hardware into separate unique virtual servers**.

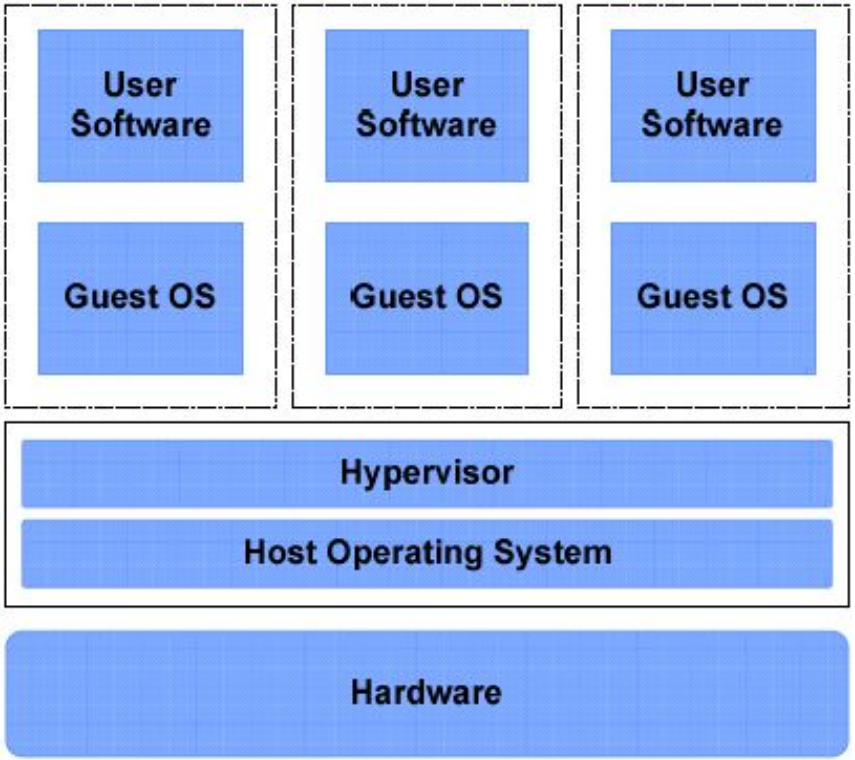
Once divided, these independent virtual servers can be **used for a multitude of tasks**.

Each **virtual server will be able to host a different operating system** without any compatibility issues.

# Types of Server Virtualization

- Full virtualization
- Para-virtualization
- Hardware-assisted virtualization
- **OS-level virtualization
- Hypervisor virtualization

# Full Virtualization



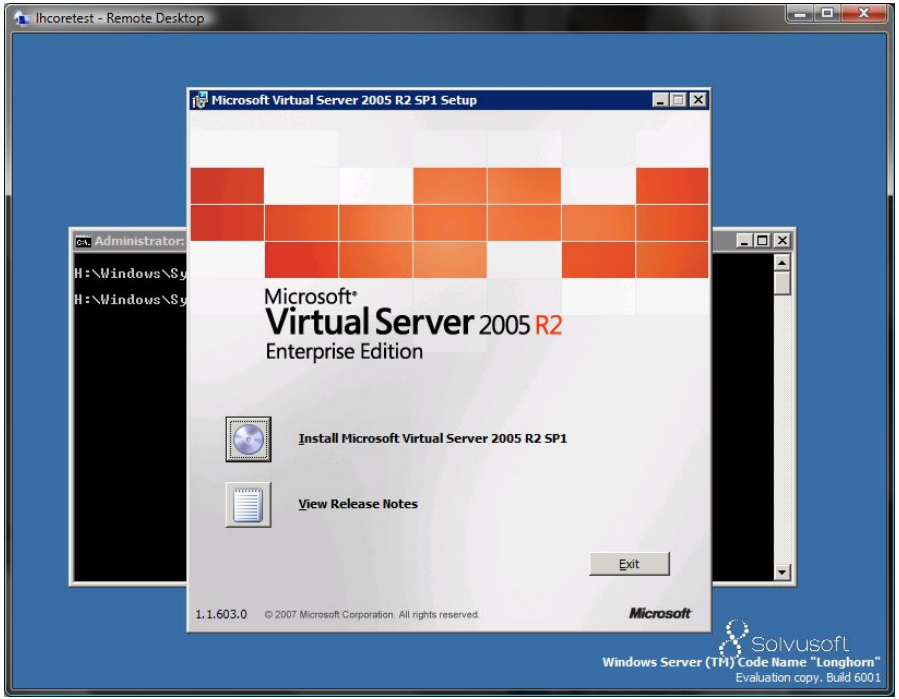
# Full Virtualization

With full virtualization, a type of software called a **hypervisor splits up the server's resources** between completely independent virtual servers that are isolated from each other.

The hypervisor **handles how resources are allocated between each virtual server**. Since the virtual machines are separate, they all run on their own operating systems and can be configured as needed.

# Full Virtualization Examples

vmware® ESXi



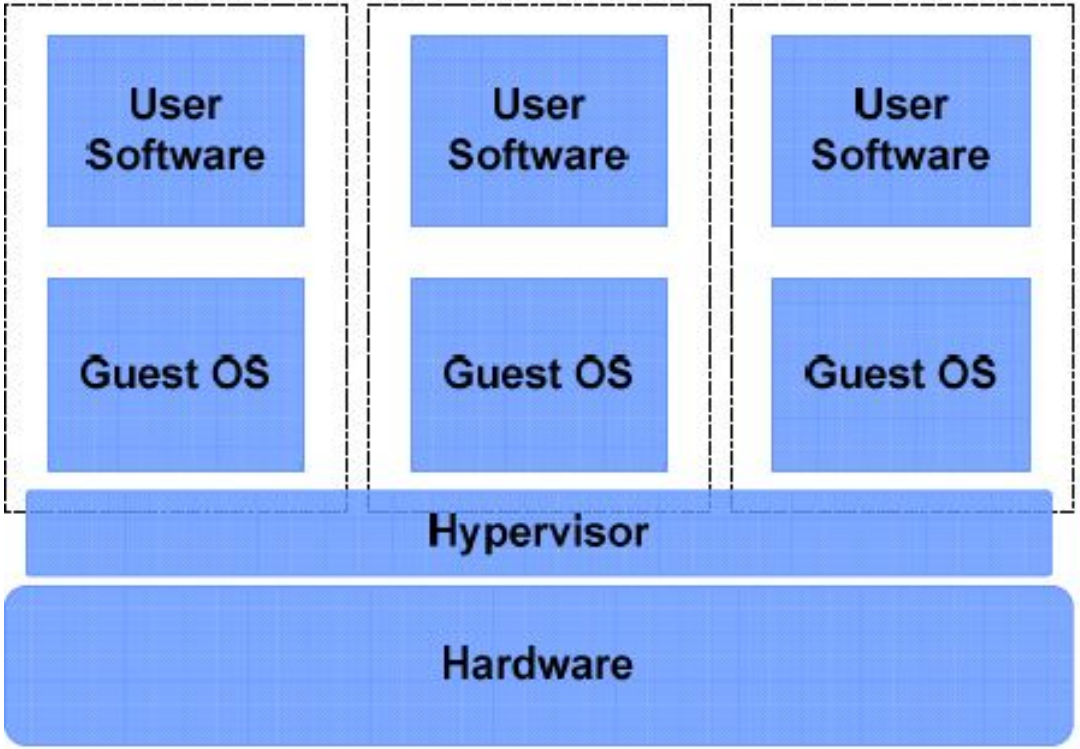
# Para Virtualization

Para-virtualization is related to full virtualization in that a hypervisor can **access virtual machines through interfaces** that are highly similar to the underlying hardware.

Prior to installation inside a virtual machine, para-virtualization involves **modifying a guest operating system** to allow all other guest OS on the server to share resources and communicate with one another.

Because all the VMs are working together, there are **fewer demands on the hypervisor, meaning more of the virtualization server's resources are dedicated to the virtual servers**.

# Para Virtualization



# Para Virtualization Examples

# Hardware-Assisted Virtualization

With hardware-assisted virtualization, the division of resources needed to support multiple VMs is already built into the CPU of the host server.

This **allows virtual machines to communicate directly to the main server rather than entirely through the hypervisor**. It's a way to partially cut out the middleman, though a hypervisor is still needed.

Since the path between the virtual machines and the physical server is more direct, the hypervisor uses a very significant amount of the server's resources.

This **makes it seem like the virtual machines are running directly on the server**.

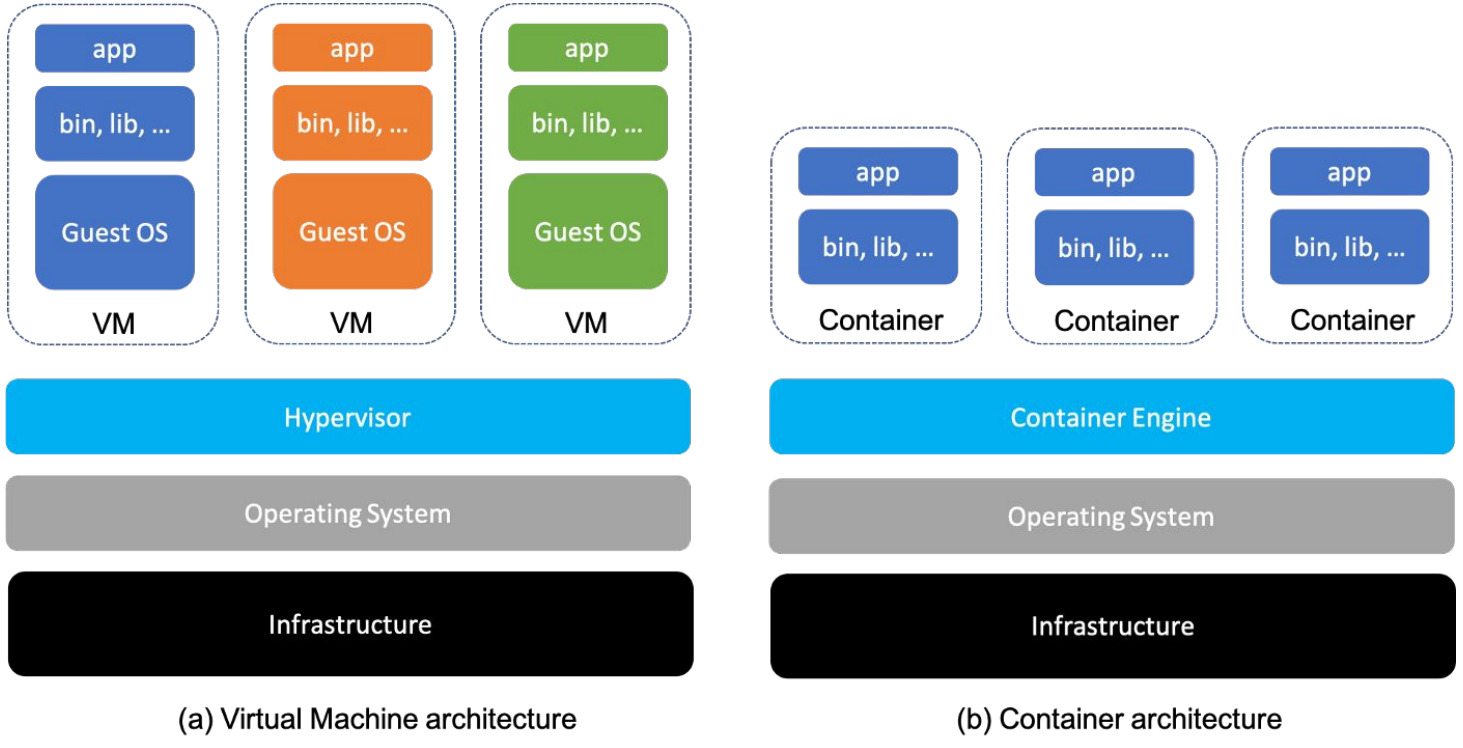
# Hardware-Assisted Virtualization Example

# OS-Level Virtualization

With full virtualization, para-virtualization, and hardware-assisted virtualization, a hypervisor is needed to provide a platform on which virtual servers' operating systems can run.

With OS-level virtualization, however, the **host server's operating system is set up to allow for multiple instances of virtual machines called containers.**

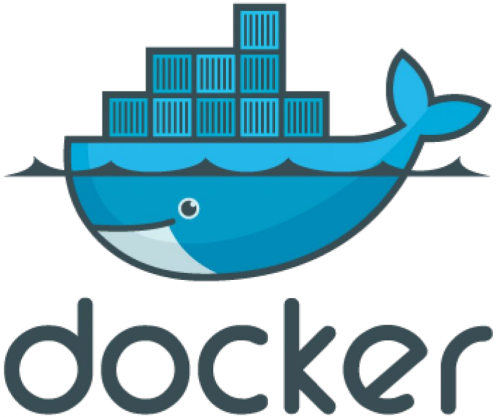
# OS-Level Virtualization




(a) Virtual Machine architecture

(b) Container architecture

# OS-Level Virtualization Examples





# Hypervisor-Based Virtualization

With hypervisor-based virtualization, software (the hypervisor) virtually emulates the hardware of the main server, basically acting like the physical machine on which operating systems can run.

The hypervisor **allocates resources of the physical server** across the various guest virtual machines.

# Types of Server Virtualization

- Full virtualization
- Para-virtualization
- Hardware-assisted virtualization
- OS-level virtualization
- Hypervisor virtualization

# Containerization

# What Is Containerization?

Containerization is a **form of virtualization**.

Virtualization aims to run multiple OS instances on a single server, whereas containerization runs a single OS instance, with multiple user spaces to isolate processes from one another.

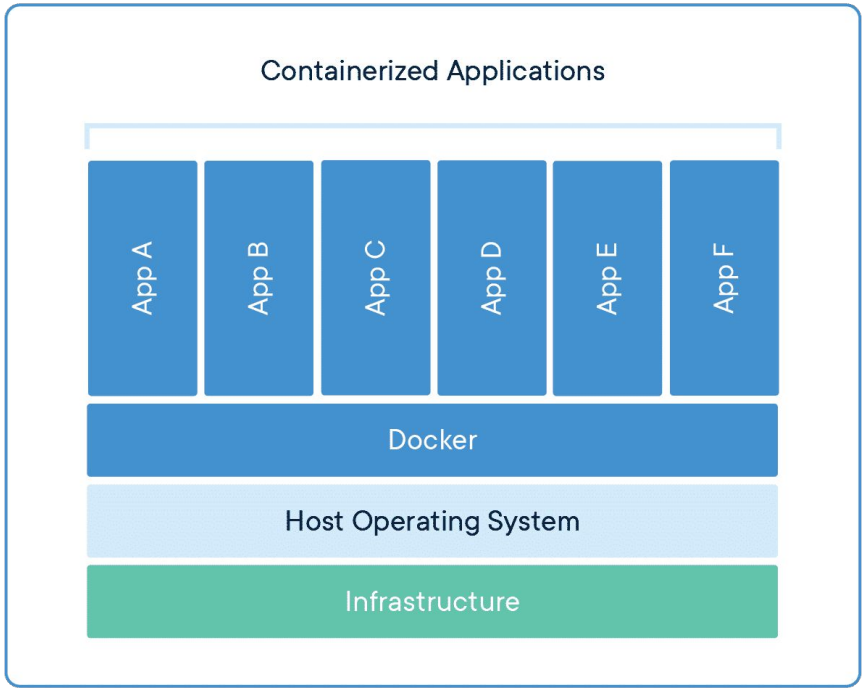
This means containerization makes sense for one AWS cloud user that plans to **run multiple processes simultaneously.**

# What Is Containerization?

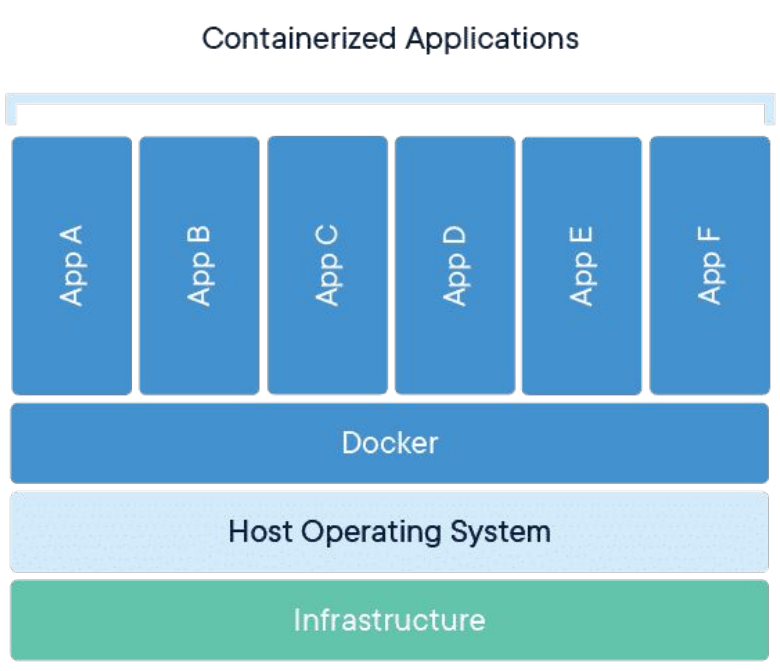
Containerization is achieved by **packaging software code, libraries, frameworks, and other dependencies together in an isolated user space called a container**.

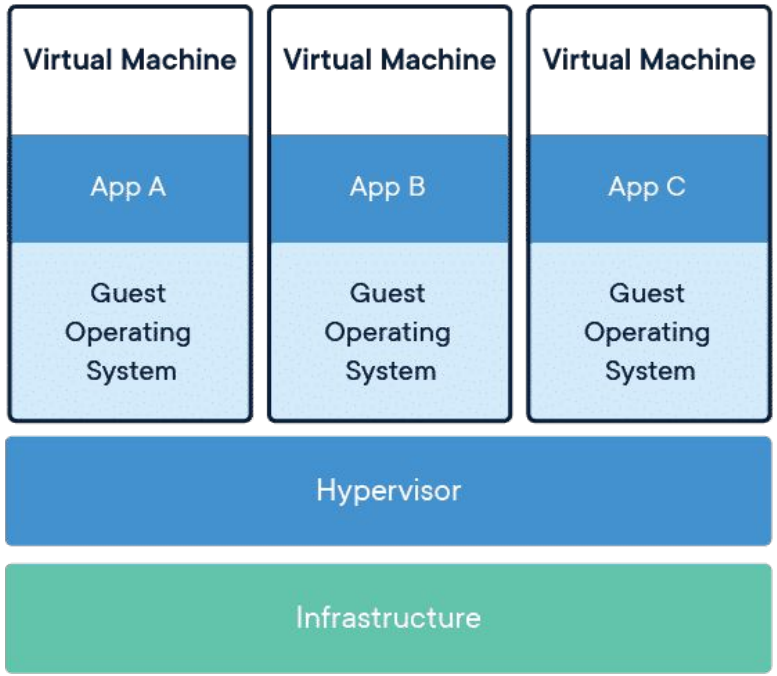
This container is **portable** and **can be used on any infrastructure in any environment** that supports the container technology, such as Docker and Kubernetes.

# Containerization



# Containerization vs Virtualization



# Containerization vs Virtualization

1. Isolation

   Virtualization results in a **fully isolated OS and VM instance**, while containerization **isolates the host operating system machine and containers from one another**. However, all containers are at risk if an attacker controls the host.

# Containerization vs Virtualization

2. Different OS

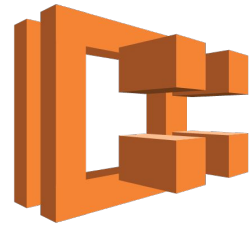
   Virtualization can **host more than one complete operating system**, each with its own kernel, whereas containerization **runs all containers via user mode on one OS**.

# Containerization vs Virtualization

3. Deployment

   Virtualization means **each virtual machine has its own hypervisor**. With containerization, either **Docker is used to deploy an individual container**, or Kubernetes is used to orchestrate multiple containers across multiple systems.

# Containerization in AWS



AWS ECS

Amazon EKS

AWS Fargate

https://aws.amazon.com/containers/

# Importance of Virtualization

# Importance of Virtualization

## Saving Space

If you’re running one task or application per server, they will soon take over your data center. By hosting multiple virtual servers on fewer physical servers, you can potentially save a lot of physical space.

# Importance of Virtualization

## Lowering Hardware Costs

Building physical servers can be costly. Virtual servers are much less expensive to deploy.

# Importance of Virtualization

## Improving Resource Efficiency

Many servers are more powerful than needed for the demands they are tasked with, meaning organizations technically pay for unused resources. Virtualization helps ensure that all resources are being efficiently used.

# Importance of Virtualization

## Lowering Energy Costs

Since virtualization allows you to efficiently use server resources rather than build unnecessary infrastructure, you have fewer servers using energy.

# Importance of Virtualization

## Decrease Demand on IT

Maintaining a large network of physical servers can be taxing on an organization’s IT department. Virtualization can help free up IT worker resources to concentrate on other needs of the business.

# Importance of Virtualization

## Speeding Up SetUp

Days or weeks can go by between purchasing hardware for physical servers and implementation. Setting up virtual servers can take minutes.

# Importance of Virtualization

## Simplifying Backups & Recovery

Backup systems on virtual machines (VMs) are quick and efficient. This means you can get up and running after a system failure fast with little to no data loss.

# What’s Next?