

# *Cloud computing*

## *TOPICS*

*1) DESTOP AS A SERVICE (DaaS)*

*2) CONTAINER AS A SERVICE (CaaS)*



# *Desktop Virtualization!*

- A technology that allows creation and storage of multiple user desktop instances on a single host, residing in a data center or the cloud
- Achieved by using a hypervisor, which resides on top of the host server hardware to manage and allow virtual desktops to utilize the computing power of the underlying server hardware.
- The hypervisor creates VMs that simulate the user's desktop environments, which can hold different operating systems, applications, personalized settings, and user data.
- Users can remotely access as well as operate these desktops from any endpoint device.



# • *Types of Desktop Virtualization*

- Desktop virtualization has two major deployment models:
- **Hosted Desktop** and **Client Virtualization**
- **Hosted Desktop Virtualization:** In this model, a server residing in a data center hosts the virtual machines. Users connect to server through standard protocols such as Remote Desktop Protocol (RDP) or connection brokers.
- There are three major variants under Hosted Desktop Virtualization:

1. **Virtual Desktop Infrastructure (VDI)**, 2. **Remote Desktop Services (RDS)**, 3.

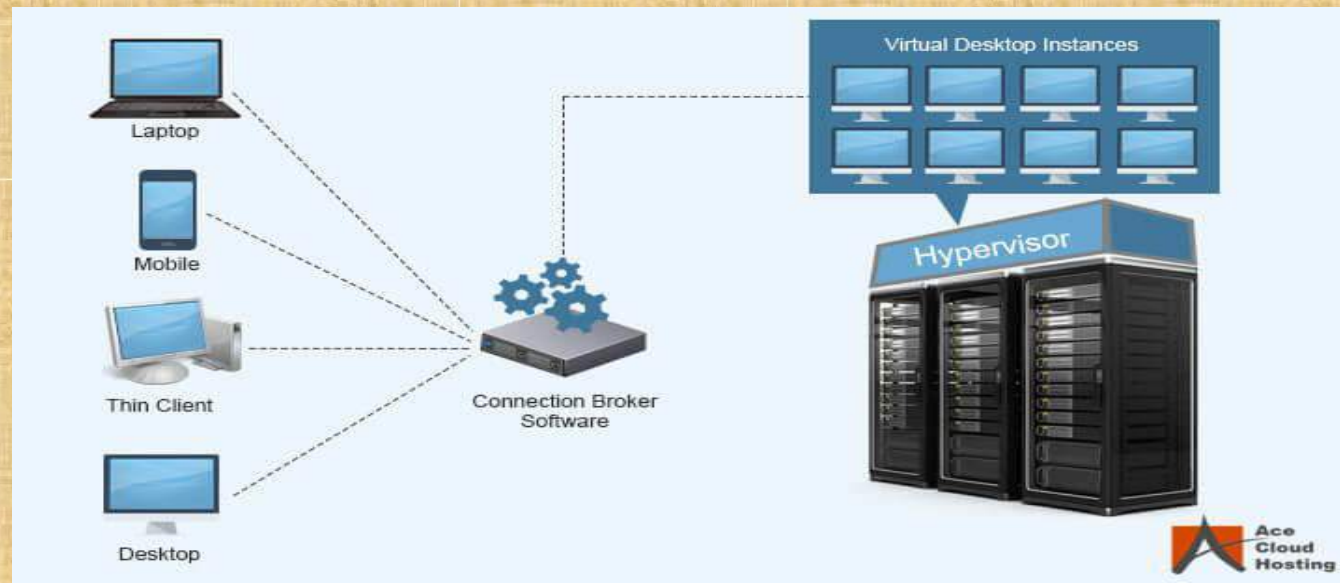
**Desktop-as-a-Service (DaaS)**

- **Client Virtualization:** In this model, a hypervisor is installed on a client device to allow us to run multiple OSes. It also has two variants:
  - 1. **Presentation virtualization**, 2. **Application virtualization**



# Virtual Desktop Infrastructure (VDI)

- In VDI, a hypervisor segments physical servers into virtual machines that in turn host virtual desktops
- Users can access these virtual desktops from any device or location
- All processing is done on the host server.
- Users connect to their desktop instances through a connection broker. Connection broker is a software-based gateway that acts as an intermediary between the user and the virtual desktop
- VDI can be either persistent or nonpersistent:
  - **Persistent VDI:** a user connects to the same desktop each time, and are able to personalize the desktop for their needs since changes (folder, cookies, cache) are saved even after the connection is reset
  - **Non-persistent VDI:** where users connect to generic desktops and no changes are saved, is usually simpler and cheaper





# *Benefits of VDI*





# *Persistent VDI Vs. Non-persistent VDI*

1. In static scenario, same user, same desktop, report to same location day after day
2. Customize and personalize
3. Stateful
4. Instances not easy to erase
5. More storage and back-up
6. Takes longer time to load machines
7. Relatively complex to patch and update
8. Expensive to run
9. Application layer settings are saved
10. Power user, Knowledge workers

1. In dynamic scenario, different user, same desktop in different session
2. Generic
3. Stateless
4. Instances easy to erase
5. No need for extra Storage
6. Quick
7. Easier to patch and update
8. Less expensive to run
9. Application layer resets at each session
10. Task group users



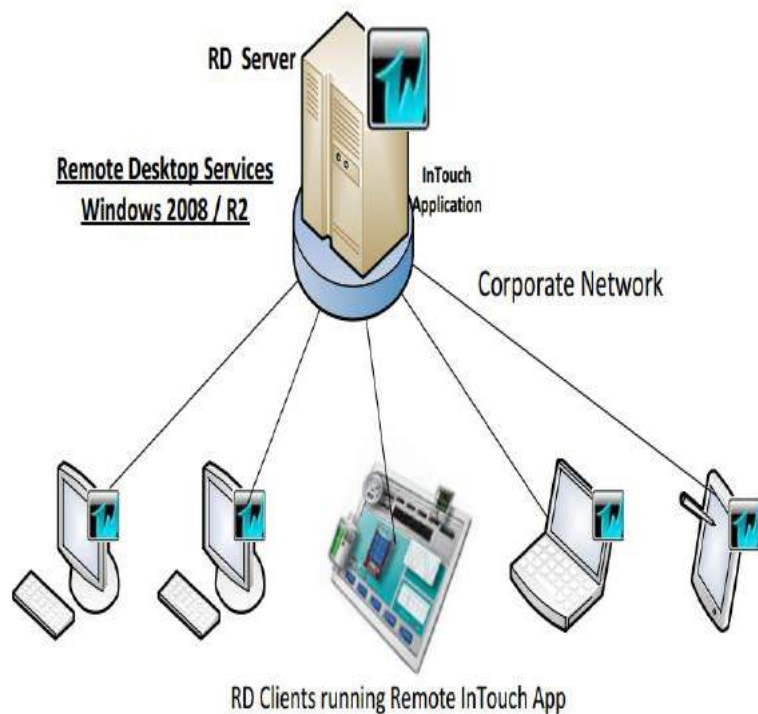
# *Remote Desktop Services (RDS)*



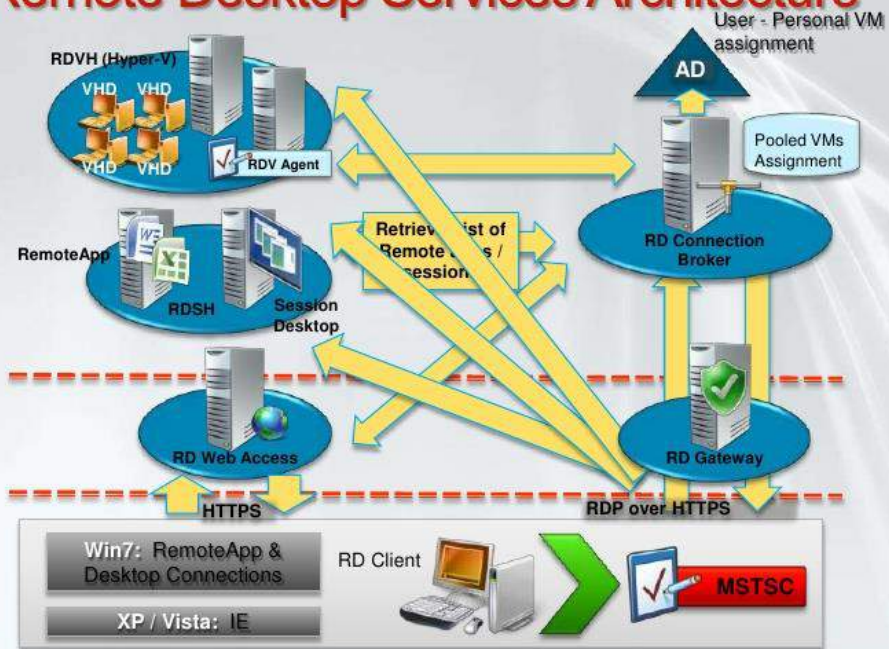
- Allows users to remotely access windows applications/ full windows desktop installed on a host server (Windows server 2008 R2 and newer) using a thin client
- One OS and set of apps (some may be incompatible) shared by multiple users
- Access from within a network or via internet (require VPN to connect securely)
- Access server via RDP & a single server can support 60+ end user
- User can't customize their desktop and, in some cases their apps.
- Quick deployment of programs, Consolidated software updates
- Remote access for user and admin, Improved performance of program
- Less expensive and less costly to implement, Single point of maintenance, only one installation for many users
- When using RDS, the desktop image you configure on the server is cloned & presented to users with all of its associated applications and data.



# RDS Services Architecture



## Remote Desktop Services Architecture





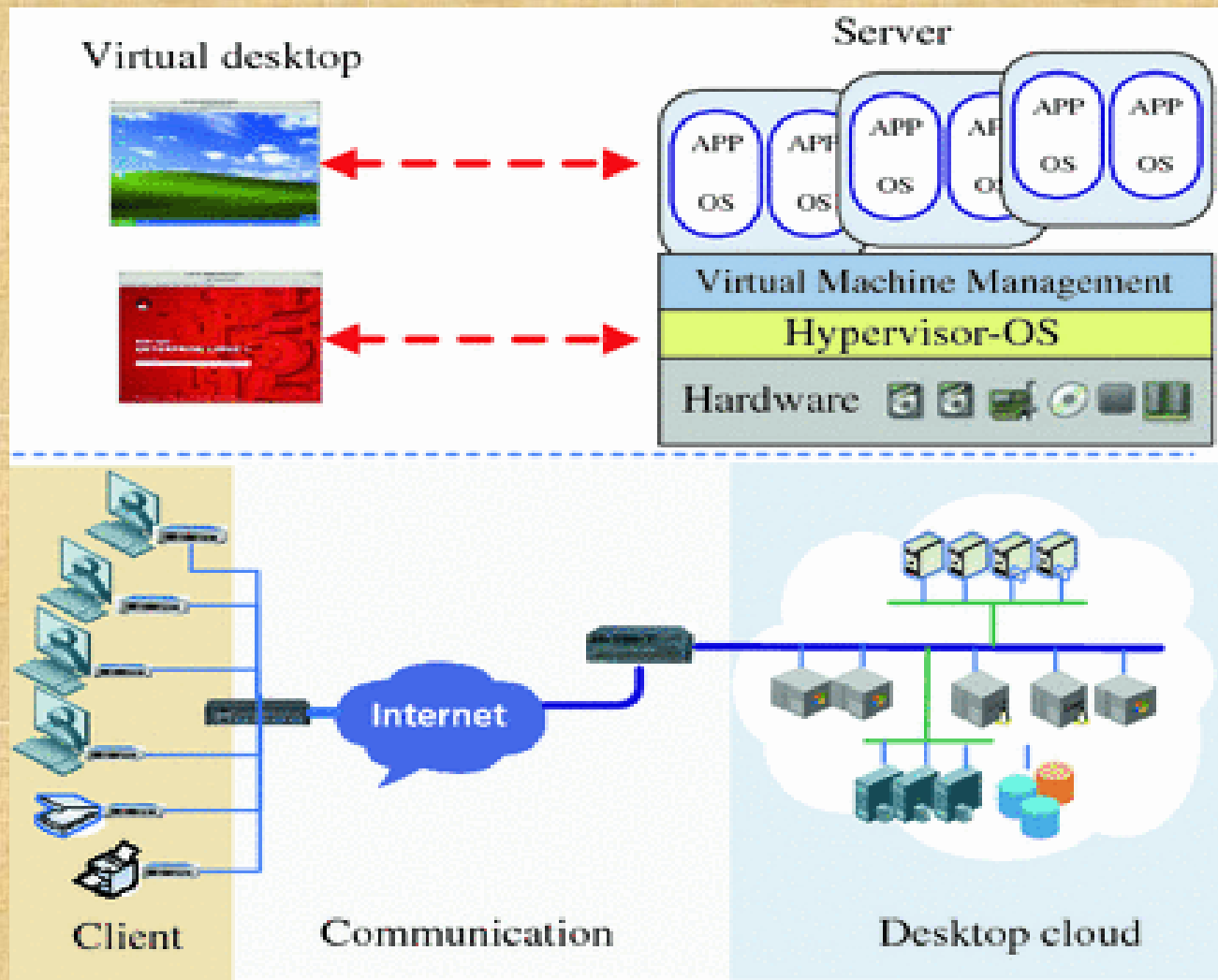
# *Desktop as a Service*



- A cloud computing offering in which third party (cloud provider) hosts (costs & maintenance) the back-end (storage, backup, security, upgrade) of VDI deployment
- One OS and set of apps (some may be incompatible) shared by multiple users
- In DaaS, Oses run inside VMs in a cloud provider's data center
- All necessary support infrastructure, including storage and network resources live in the cloud
- End users may access them through client software or a web browser
- Quick deployment of programs, Consolidated software updates
- Remote access for user and admin, Improved performance of program
- Less expensive and less costly to implement, Single point of maintenance, only one installation for many users
- When using RDS, the desktop image you configure on the server is cloned & presented to users with all of its associated applications and data.



# *DaaS Architecture*





# DaaS Vendor Comparisons



PROVIDER	SUPPORTED GUEST OSES	WINDOWS DELIVERY	STARTING PRICE	FREE TIER OR TRIAL	SUPPORT
Amazon WorkSpaces	Windows and Linux	Server-based	\$21/month	Free tier for 12 months	Multiple support tiers through AWS
Citrix Managed Desktops	Windows	Desktop and server-based	\$8/month	Request 14-day free trial	Multiple support tiers through Citrix
Cloudalize DaaS	Windows	Server-based	\$1/hour + \$20/month	No	Contact vendor for support info
DinCloud DinWorkspace	Windows	Server-based	\$50/month	No	24/7 support
Evolve IP Workspaces	Windows	Server-based	Contact vendor	No	Phone and email support
Microsoft Windows Virtual Desktop	Windows and Linux	Desktop and server-based	Free to eligible customers	Free Azure services for 12 months	Multiple support tiers through Azure
VMware Horizon Cloud	Windows and Linux	Desktop and server-based	\$8/month	Free 90-day trial	Multiple support tiers through VMware



## VDI vs. RDS vs. DaaS

	Virtual Desktop Infrastructure (VDI)	Remote Desktop Services (RDS)	Desktop as a Service (DaaS)
HOSTING METHODS	Desktops and applications hosted on physical servers by an organization, in-house	Desktop and application sessions hosted on a shared desktop on Windows Server	Desktops and applications hosted in the cloud by a third-party vendor
REQUIREMENTS	Requires organizations to build out their own virtualization hardware and servers	Requires a Windows Server virtual desktop environment	Requires no server or data center investment
LICENSING	Vendors offer different license models based on persistent or nonpersistent desktops on a per-user or per-machine basis	Requires a client access license for each unique user that establishes a connection with the RDS host	With public cloud-hosted DaaS, the vendor takes care of the licensing as part of the fees. However, private cloud-hosted DaaS can be very complicated.
RESOURCES	High amount of labor and resources required for setup	Medium amount of labor and resources to set up Windows Server and RDS	Low amount of resources and labor required for setup



# *Client Virtualization*

- **A virtual machine environment in the user's machine**
- **User's computer hosts multiple VMs**
- **Each of which contains an OS and set of applications**
- **Virtual PC, parallel desktop, VMware fusion and Virtual Box**
- **Compatibility to run different OSes or different versions of the same OS**
- **Support an heterogeneous environment and**
- **Provide maximum productivity by allowing a single user to multi-task with VMs**



# *Container as A Service (CaaS)?*



- Allows users to upload, organize, start, stop, scale, and maintain containers, applications and clusters
- Enables these processes using either a container-based virtualization, an API, or a web portal interface
- It helps users construct security-rich, scalable containerized applications through on-premises data centers or the cloud
- Containers and clusters are used as a Service, and are deployed in the cloud or in on-site data centers

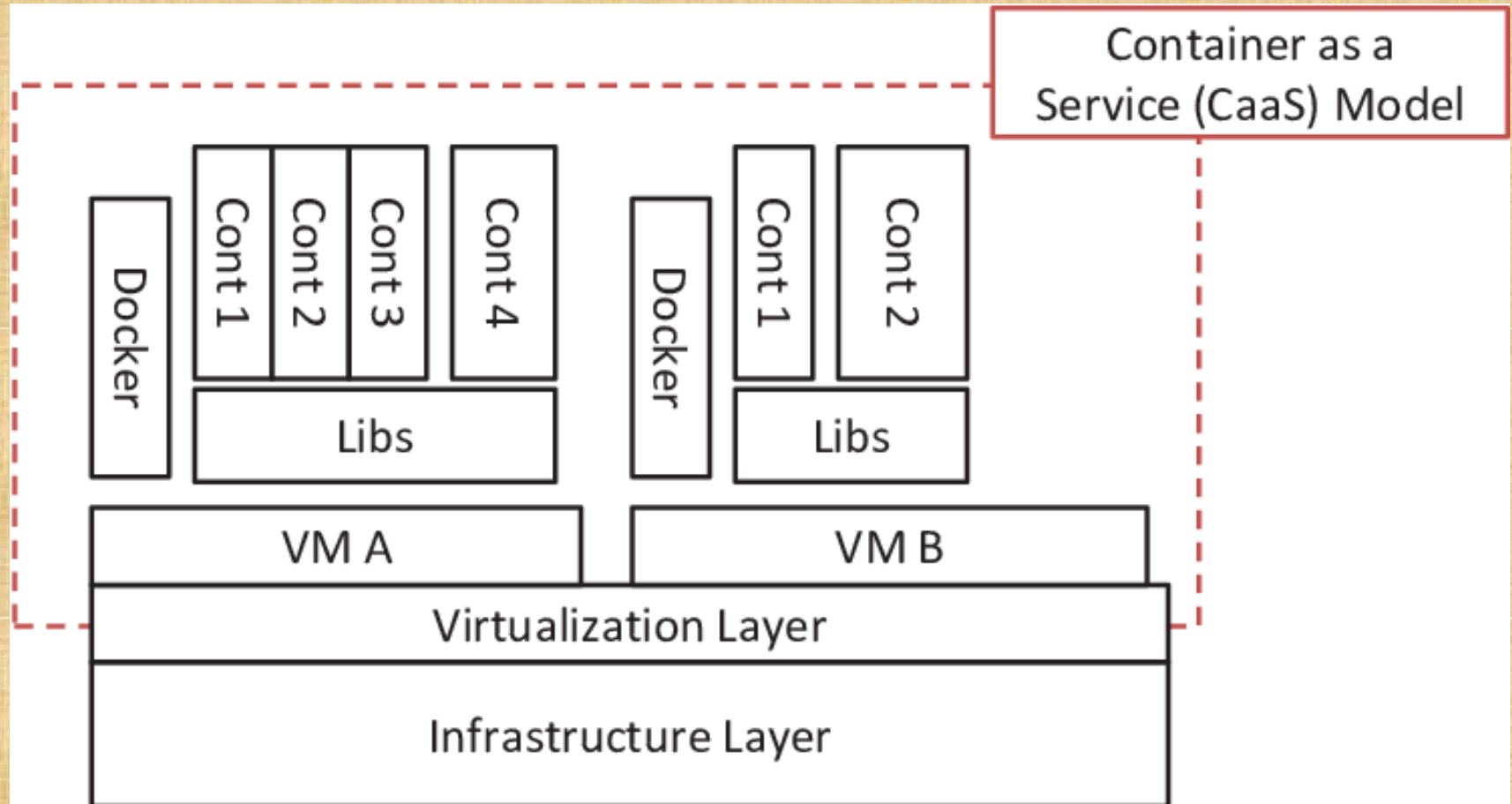


# *Container?*

- An executable unit of software in which an application code is packaged along with its libraries and dependencies
- Such that it can be run anywhere, whether it be on desktop, traditional IT, or the cloud
- Takes advantage of OS virtualization in which features of OS are leveraged to both isolate processes and
- And control the amount of CPU, memory, and disk that those processes have to.



# Container as A Service?





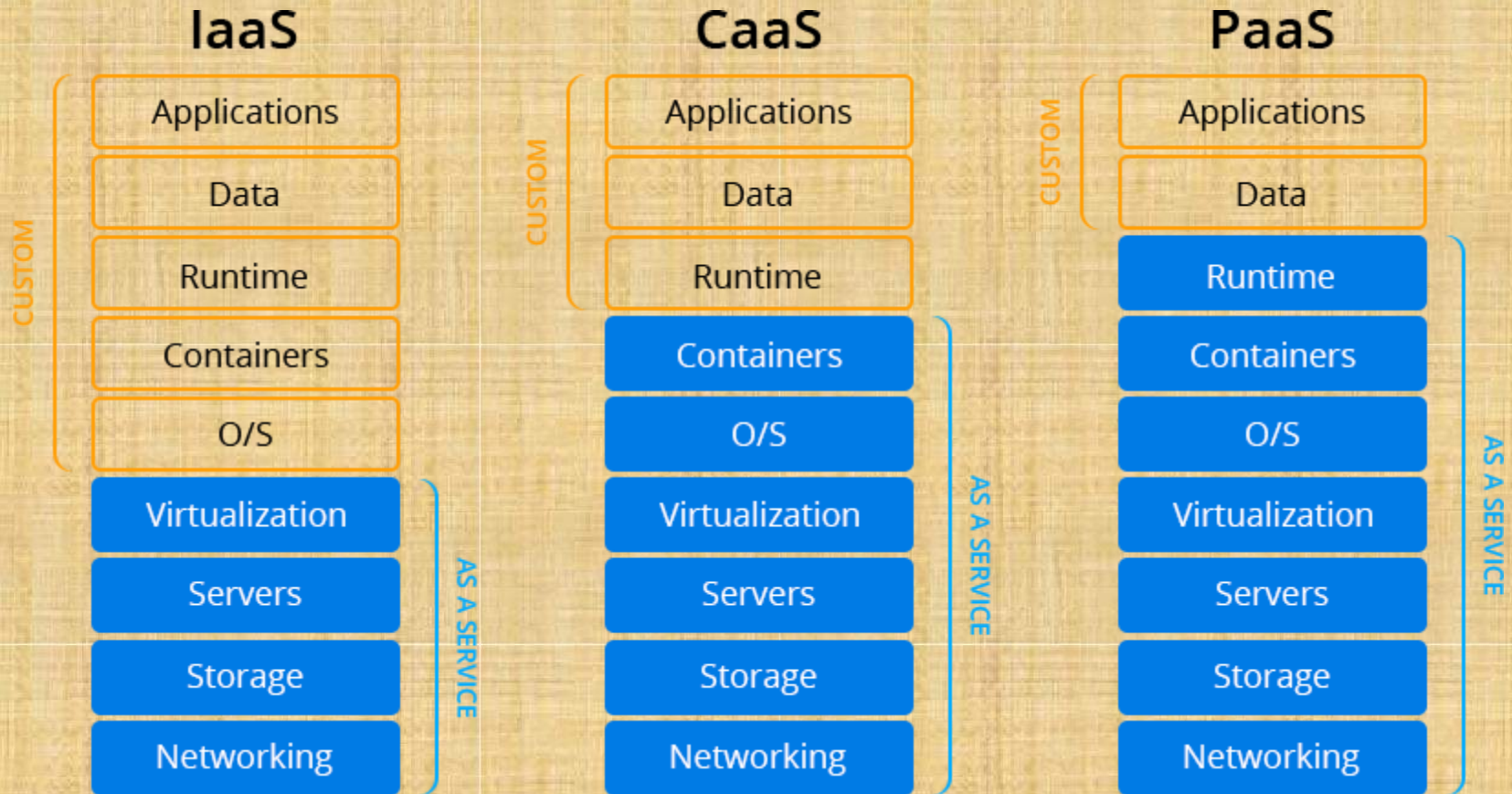
# *Container VS. Virtual Machine*

- The advantage as compared to Virtual Machines on VMWare is that individual instances don't each require an operating system.
- The footprint is much lower, and new instances can be started in seconds rather than in minutes.
- So whenever an instance fails, a new instance will be spawned within seconds, which is noticeable in most use-cases.





# IaaS VS. CaaS VS. PaaS





# Platform Independent Services

## CONTAINER-AS-A-SERVICE - THE NEW PARADIGM



### IMPORTANT FACTS

Based on Docker open-source software, isolating applications using OS virtualization

Established in 2013, now over 100,000 Docker-based applications

Quick and straightforward distribution thanks to the "Build once, run everywhere" principle

T...Systems



# *Why Container Important?*

- When an application is created in a container, that completed app. Has everything it needs to run, including dependencies and configuration files
- So, easy to reliably launch application in different environment, public or private cloud
- This portability grants enterprises a large amount of flexibility, accelerating development process
- Making app. Easier to switch to a different provider or cloud environment
- Highly efficient & cost-cutting as container does not need a separate OS and requires less resources than a VM




# Contd..

















- Several container can be run on a single server
- Containers don't interact and are isolated from other container, although they do share the same resources
- If an application crashes in one container, other container can continue to use without any issue (security)
- Speedily launch updates and security patches
- Since, container does not OS, so takes second to create, start, replicate or destroy a container
- Allowing end user to incorporate multiple identical containers within same cluster to scale out
- Have an effective & efficient development pipeline
- Takes little time to modify configuration file to generate new containers and delete previous one



## Four Common Container Solution Models

 Public Cloud Vendor

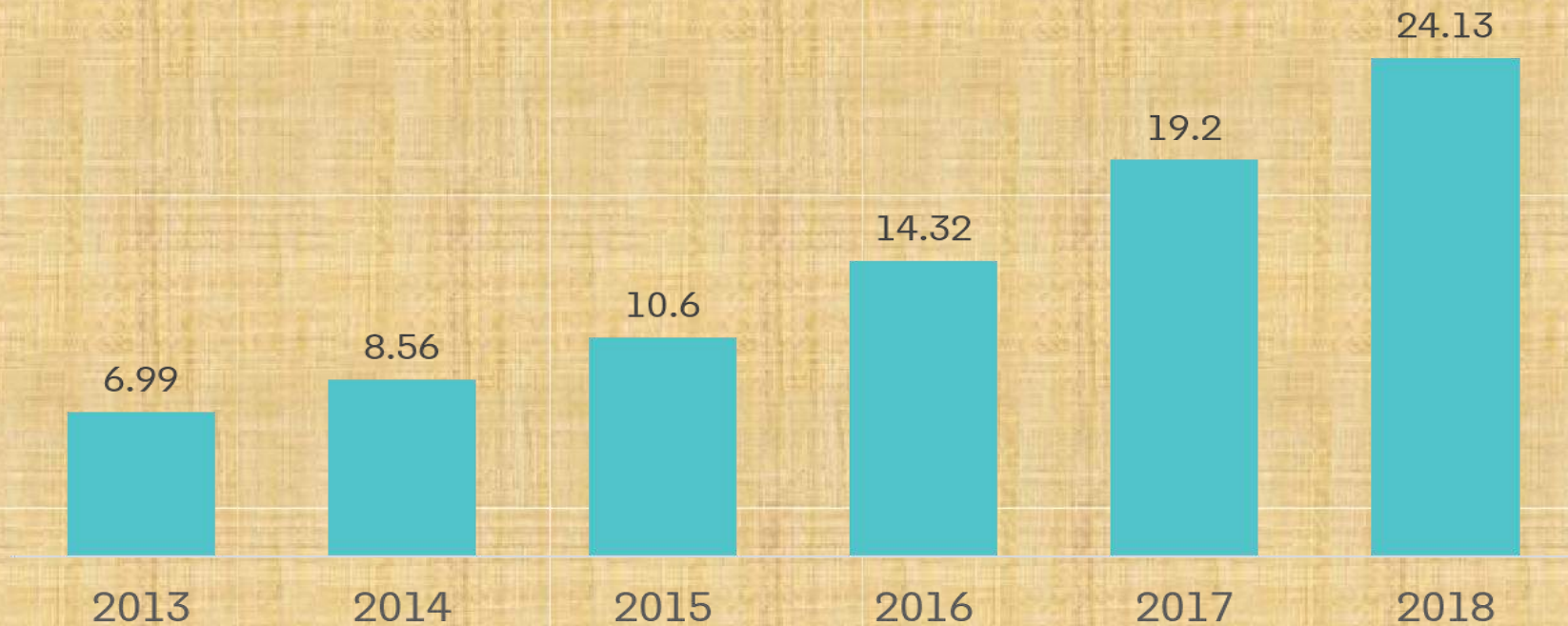
 Third-party service provider / enterprise IT

Solution models	Examples	Management of physical infrastructure	Management of container orchestration platform	Management of containers
 Private cloud containers	OpenShift, GKE, ICP, on premise			
 Containers hosted on public cloud	OpenShift, GKE, ICP, on public cloud			
 Public cloud containers	AWS EKS, Azure AKS			
 Container-as-a-service	AWS Fargate, Azure Container Instances			

 Everest Group® Pricing Services



## Volume of Digital Payment Transactions, in millions, India, 2013 - 2018



Source: RBI

