BITS Pilani

Cloud Computing

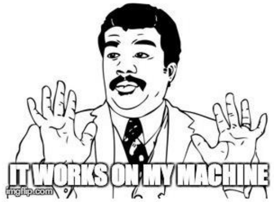
SEWP ZG527

Agenda

- Container Recap
- Introduction to PaaS
- Building blocks of PaaS
- Characteristics of PaaS
- Advantages and Risks
- PaaS Example – Windows Azure

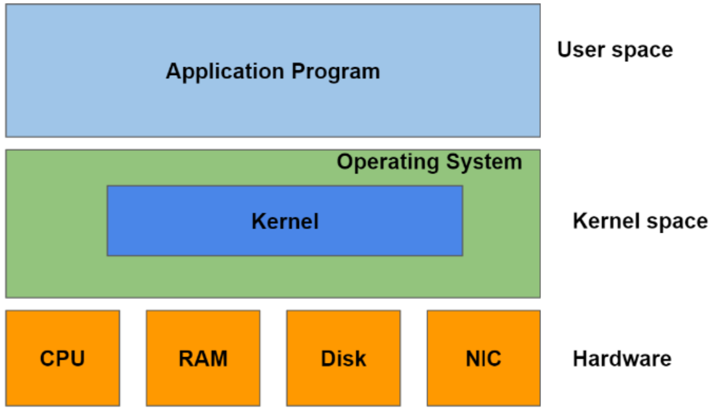
Recap

Motivation towards Containers



It works on my machine

- Let’s assume that you are building a web service on an Ubuntu machine. Your code works fine on your local machine. You have a remote server in your data centre that can run your application.
- You copy your local binaries on the remote server and try to run your code. The next thing that you see is your code doesn’t work there. The above problems result in portability issues. The developer has to spend a lot of time debugging the environment-specific issues.



Working of an Operating System

- The computer has different hardware resources such as RAM, hard disk, Network Interface Card, IO devices, etc. The operating system is the software that manages this hardware.
- OS consists of a system program known as the kernel, which is loaded in the memory when OS starts. The kernel is responsible for process management, CPU scheduling, file system & IO.
- User programs interact with the hardware through the means of the kernel. For eg:- Let’s say your application wants to open a file and write content in it. The application will invoke system calls like *fopen*() and *fwrite*() to perform its functions.
- The kernel performs the function on behalf of the user program and gives the output back to it. The following diagram shows the different layers involved in the functioning of an application program.

What are Containers?

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. The way which containerized applications operate is shown ➔

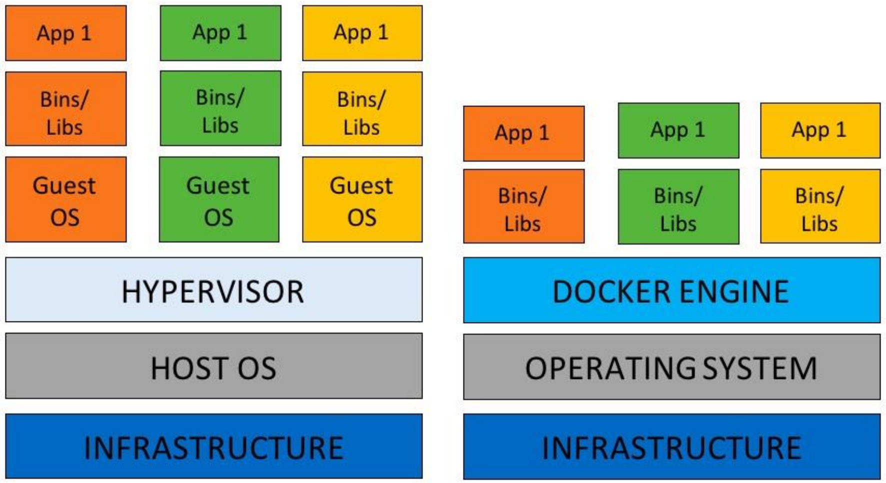
Containers are an operating system virtualization technology used to package applications and their dependencies and run them in isolated environments.

They provide a lightweight method of packaging and deploying applications in a standardized way across many different types of infrastructure

Containers run consistently on any container-capable host, so developers can test the same software locally that they will later deploy to full production environments.

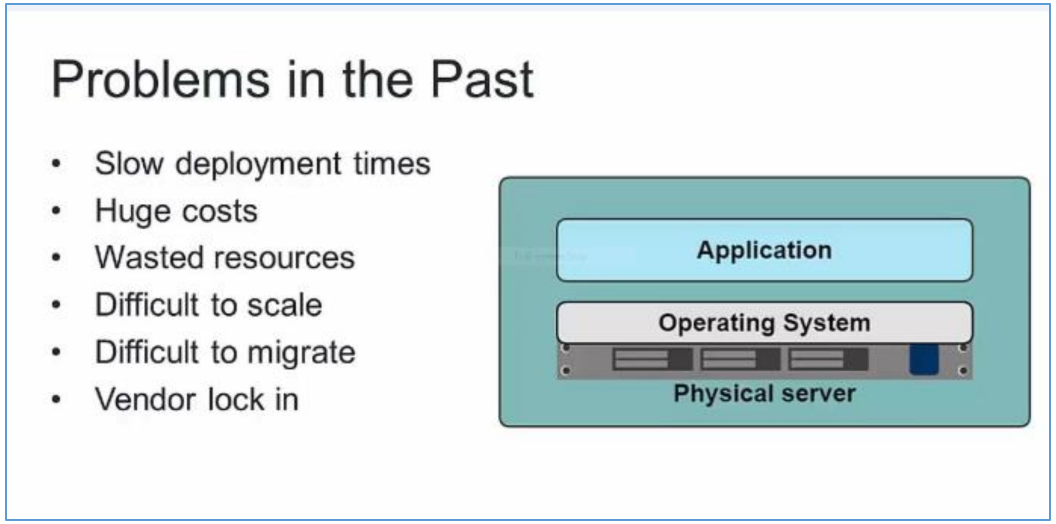
The container format also ensures that the application dependencies are baked into the image itself, simplifying the hand off and release processes.

Because the hosts and platforms that run containers are generic, infrastructure management for container-based systems can be standardized.



As can be seen from this diagram, a container includes an application plus any binaries or libraries that the application requires in order for it to run.
The container runs under the control of the container engine (such as Docker or CRI-O), which in turn runs on top of the operating system (which can be Windows 10, Windows Server 2016, or Linux depending on the container engine being used).

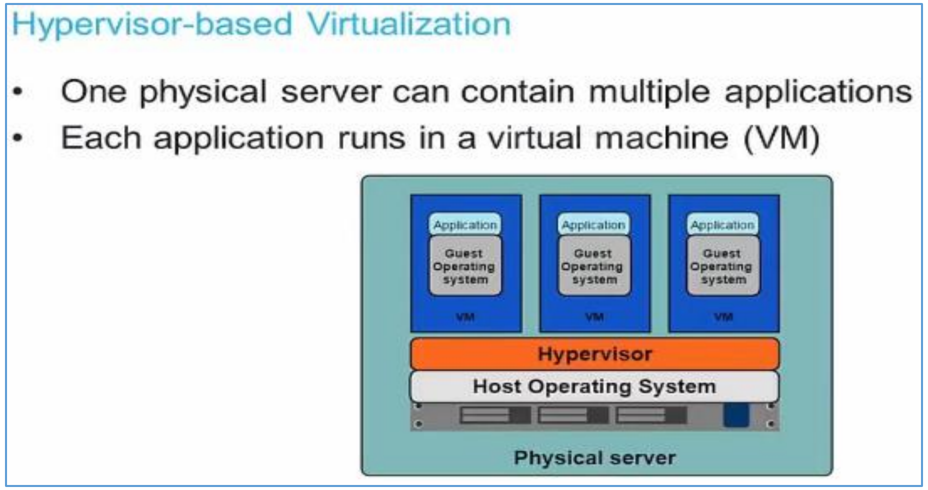
Dockers - Motivation



Problems in the Past

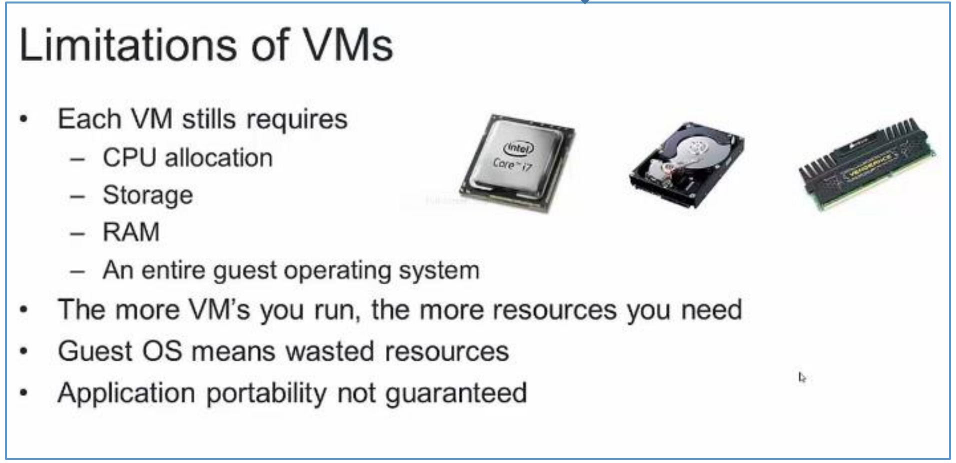
- Slow deployment times
- Huge costs
- Wasted resources
- Difficult to scale
- Difficult to migrate
- Vendor lock in

Virtualization



Hypervisor-based Virtualization

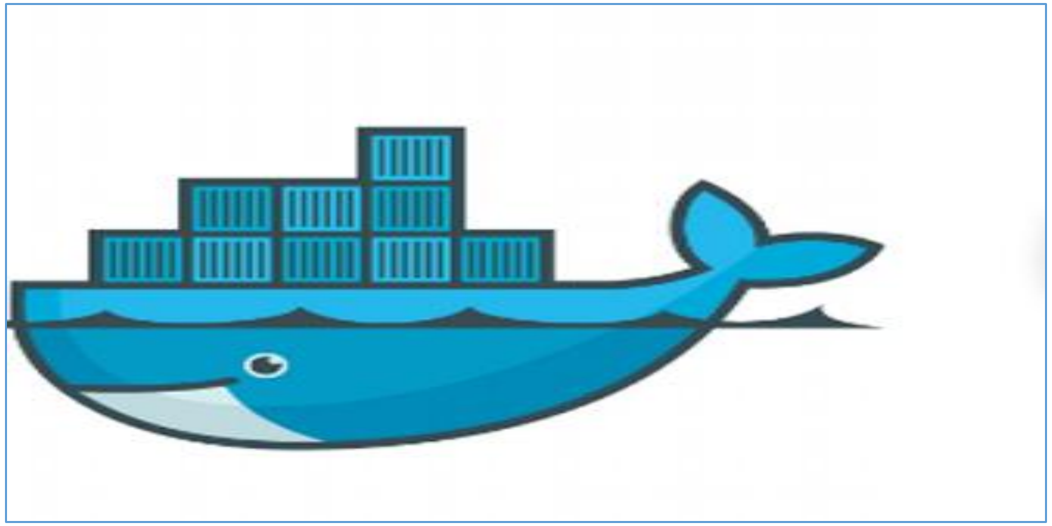
- One physical server can contain multiple applications
- Each application runs in a virtual machine (VM)



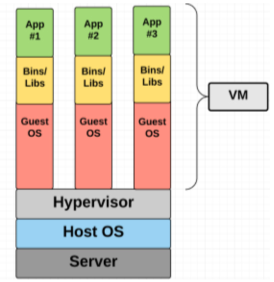
Limitations of VMs

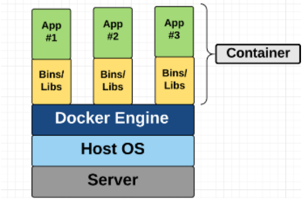
- Each VM stills requires
  - CPU allocation
  - Storage
  - RAM
  - An entire guest operating system
- The more VM's you run, the more resources you need
- Guest OS means wasted resources
- Application portability not guaranteed

Containers



Difference between VM & Container



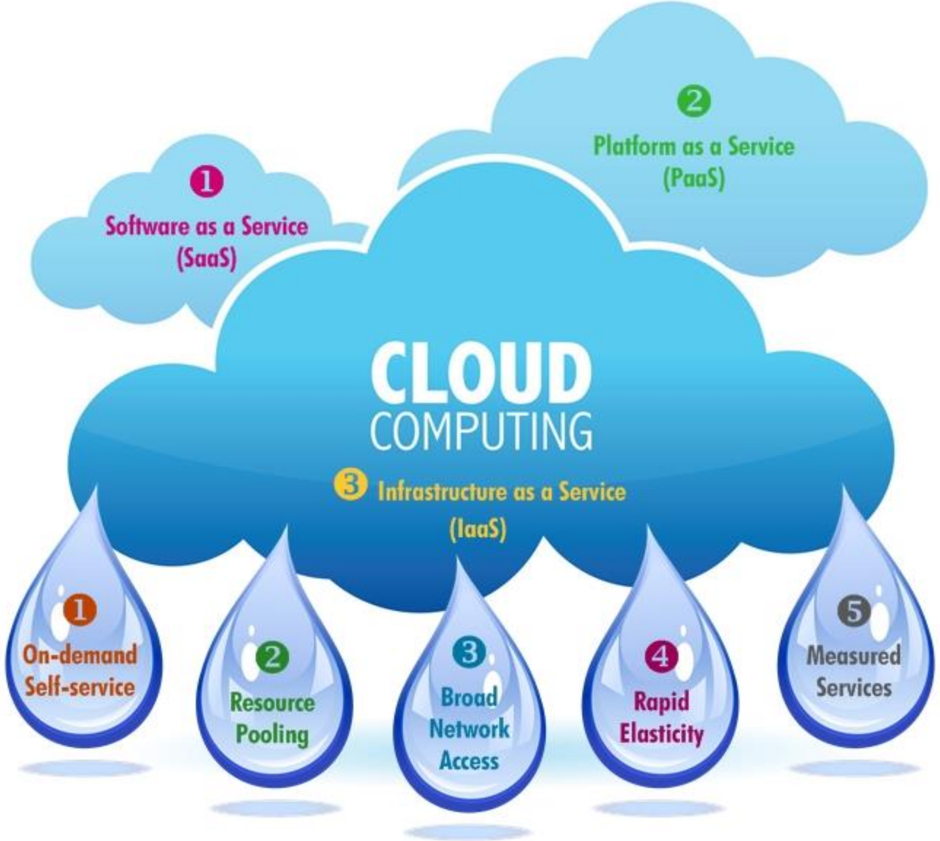


VM	Container
Virtual machines, or VMs, are a **hardware virtualization technology** that allows you to fully virtualize the **hardware and resources of a computer**.	**Containers** take a different approach. **Rather than virtualizing** the entire computer, **containers virtualize the operating system directly**.
A **separate guest operating system** manages the **virtual machine**, separate from the OS running on the host system.	They run as **specialized processes managed by the host operating system's kernel**, but with a **constrained and heavily manipulated** view of the system's processes, resources, and environment.
On the **host system**, a piece of software called a **hypervisor** is responsible for starting, stopping, and **managing** the virtual machines.	**Containers** are **unaware** that they exist on a **shared system** and **operate** as if they were in **full control of the computer**.
Because VMs are operated as completely distinct computers that, under normal operating conditions, cannot affect the host system or other VMs, virtual machines offer great isolation and security.	it is more common to manage containers more similarly to applications.
In general, virtual machines let you subdivide a machine's resources into smaller, individual computers, but the result doesn't differ significantly from managing a fleet of physical computers.	containers occupy a space that sits somewhere in between the strong isolation of virtual machines and the native management of conventional processes.

innovate achieve lead

BITS Pilani

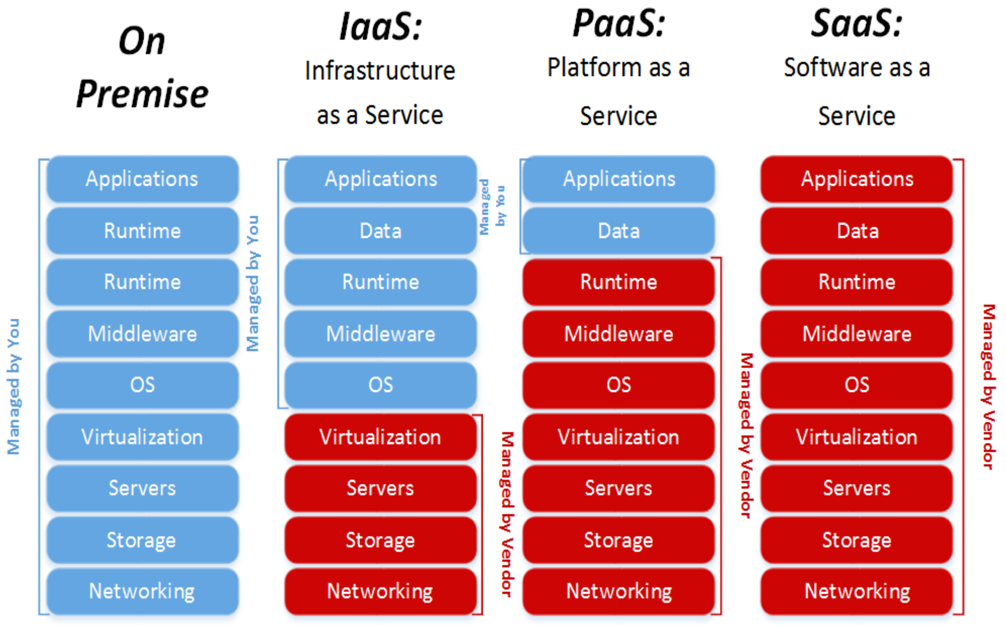
Pilani|Dubai|Goa|Hyderabad



Platform as a Service
PaaS

Introducing Platform as a Service

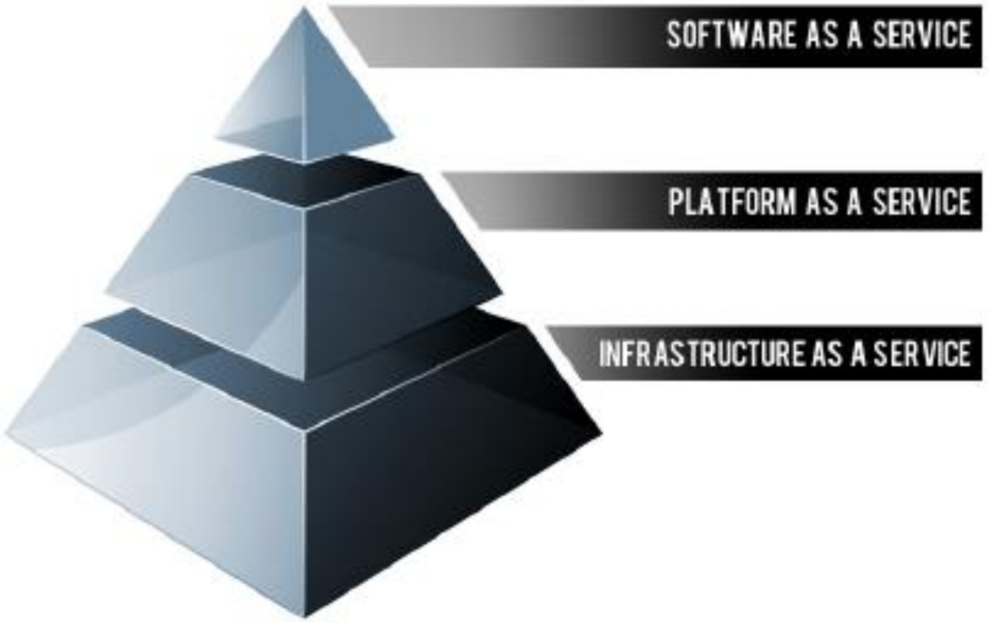
- The capability provided to the consumer is to deploy onto the cloud infrastructure, consumer-created or acquired applications created using programming languages and tools supported by the provider.
- The consumer **does not manage** or control the underlying **cloud infrastructure** including **network**, **servers**, **operating systems**, or **storage**, but has control over the **deployed applications** and possibly application **hosting** environment **configurations**.
- A **PaaS platform** offers an environment on which developers **create** and deploy **applications** and do not **necessarily** need to **know** how many **processors** or how much **memory** that **applications** will be **using**.
- In addition, multiple programming models and **specialized services** (e.g., data access, authentication, and payments) are offered as building blocks to new applications.
- Google AppEngine, Azure, Force.com an example of Platform as a Service



On Premise	*IaaS:* Infrastructure as a Service	*PaaS:* Platform as a Service	*SaaS:* Software as a Service
Applications	Applications	Applications	Applications
Runtime	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Building Blocks - PaaS

- PaaS providers can assist developers from the conception of their original ideas to the creation of applications, and through to testing and deployment.
- Below are some of the features that can be included with a PaaS offering:
    - ❑ Operating system
    - ❑ Server-side scripting environment
    - ❑ Database management system
    - ❑ Server Software
    - ❑ Support
    - ❑ Storage
    - ❑ Network access
    - ❑ Tools for design and development
    - ❑ Hosting



Characteristics- PaaS

- Services to develop, test, deploy, host and maintain applications in the same integrated development environment. All the varying services needed to fulfill the application development process

- Web based user interface creation tools help to create, modify, test and deploy different UI scenarios

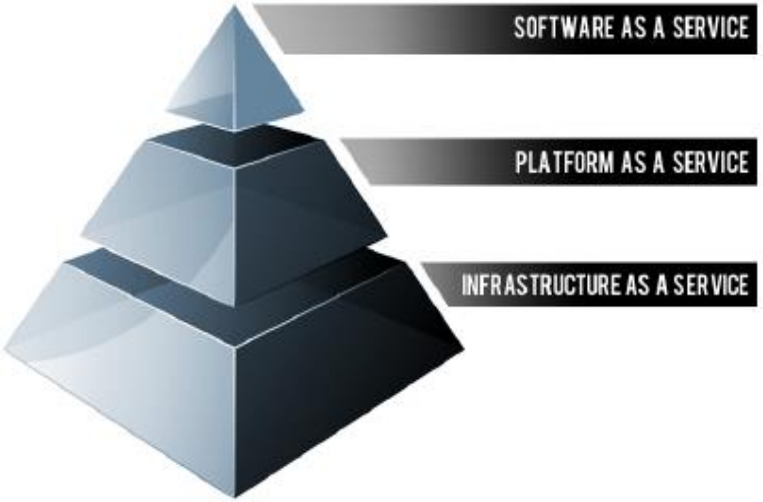
- Multi-tenant architecture where multiple concurrent users utilize the same development application

- Built in scalability of deployed software including load balancing and failover

- Integration with web services and databases via common standards

- Support for development team collaboration – some PaaS solutions include project planning and communication tools

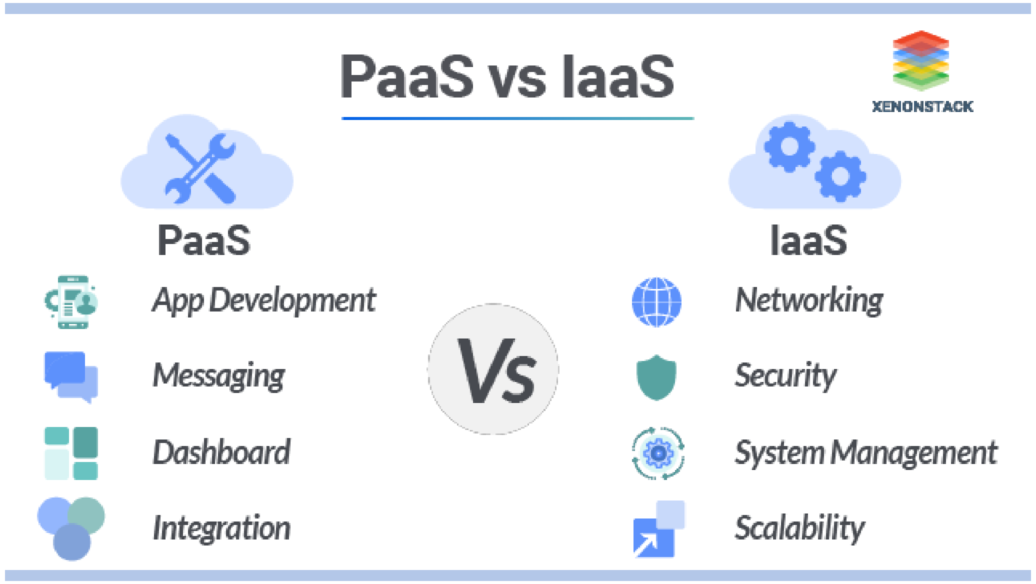
- Tools to handle billing and subscription management



PaaS vs IaaS

PaaS, which is similar in many ways to Infrastructure as a Service, is differentiated from IaaS by the addition of value added services and comes in two distinct flavours;

1. A collaborative platform for software development, focused on workflow management regardless of the data source being used for the application. An example of this approach would be Heroku, a PaaS that utilizes the Ruby on Rails development language.
2. A platform that allows for the creation of software utilizing proprietary data from an application. This sort of PaaS can be seen as a method to create applications with a common data form or type. An example of this sort of platform would be the Force.com. PaaS from Salesforce.com which is used almost exclusively to develop applications that work with the Salesforce.com CRM



PaaS Advantages

Advantages

•Users don’t have to invest in physical infrastructure

• PaaS allows developers to frequently change or upgrade operating system features. It also helps development teams collaborate on projects.

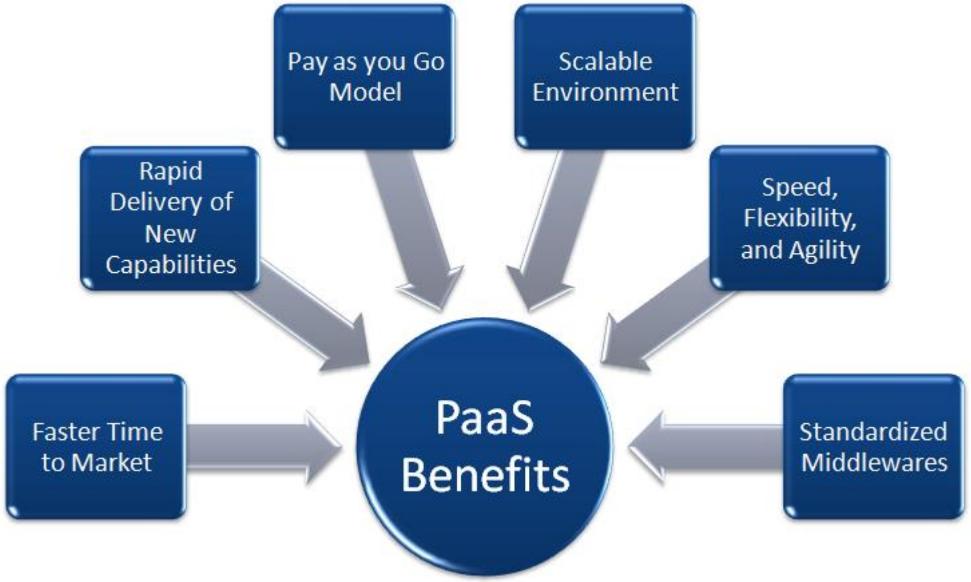
•Makes development possible for ‘non-experts’

•Teams in various locations can work together

•Security is provided, including data security and backup and recovery.

•Adaptability; Features can be changed if circumstances dictate that they should.

•Flexibility; customers can have control over the tools that are installed within their platforms and can create a platform that suits their specific requirements. They can ‘pick and choose’ the features they feel are necessary.



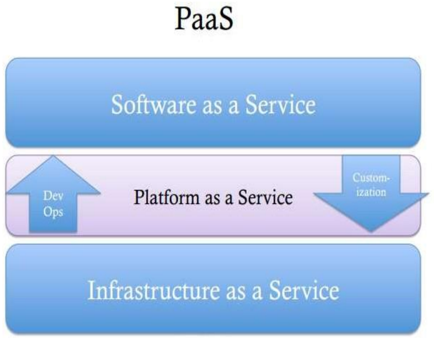
PaaS Disadvantages

- Since users rely on a provider's infrastructure and software, vendor lock-in can be an issue in PaaS environments.
- Other risks associated with PaaS are provider downtime or a provider changing its development roadmap.
- If a provider stops supporting a certain programming language, users may be forced to change their programming language, or the provider itself. Both are difficult and disruptive steps.

Who Can Use PaaS

Developers

- My code – running
  - Not a “VM” but a Virtual App Server
- Not just code
  - I like Queues and Topics, ESB flows, Workflows, Databases, Logs, Portals, etc.
- Not just Runtime
  - I like SVN, Git, build, continuous integration, code coverage, automated test



Business

- Moreover, if you are a manager of a group of developers, you probably like governance.

- Software developers, web developers and businesses can benefit from PaaS.
- **For example,** web developers can use individual PaaS environments at every stage of the process to develop, test and ultimately host their websites. However, businesses that are developing their own internal software can also utilise Platform as a Service, particularly to create distinct ring-fenced development and testing environments.

PaaS – Best Practices

Managing PaaS

- If you choose to maintain the software yourself, you must set up, configure, maintain, and administer the PaaS yourself (either on a public or private cloud).
- Alternatively, you can have the vendor to provide these services. The result is reduced friction between the development and deployment teams. There will, of course, be situations in which it’s critical for the internal team to control and manage a complex software environment.

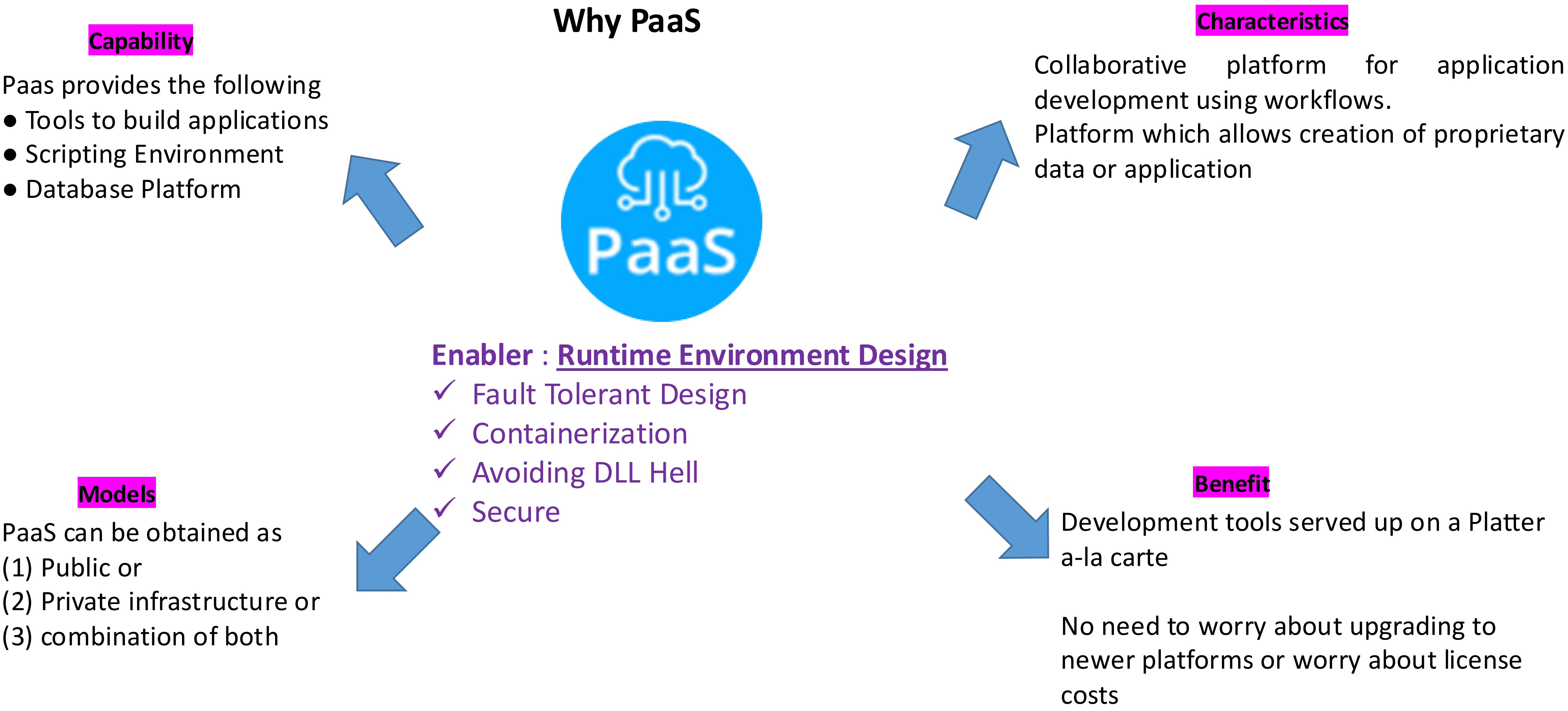
Best Practices

- Start with the data, and work up to the services and UI. No matter what the PaaS provider suggests.
- Define a staging and testing strategy before you begin development.
- Consider SOA approaches in the design and deployment of the PaaS- bases application.
- Make sure to do load testing along with functional testing.
- Make sure to model performance.
- Don’t fall in love with a PaaS player, you may need to use several.

PaaS – AWS Examples

- AWS Lambda
  - Serverless code
  - On demand / respond to events
- AWS Elastic Beanstalk
  - Focus on developing features of your web application
  - Amazon takes care of provisioning resources, scaling, patching etc.

Platform as a Service - Summary



Why PaaS

Capability

Paas provides the following
- Tools to build applications
- Scripting Environment
- Database Platform

Characteristics

Collaborative platform for application development using workflows.
Platform which allows creation of proprietary data or application

Enabler : **Runtime Environment Design**

✓ Fault Tolerant Design
✓ Containerization
✓ Avoiding DLL Hell
✓ Secure

Models

PaaS can be obtained as
(1) Public or
(2) Private infrastructure or
(3) combination of both

Benefit

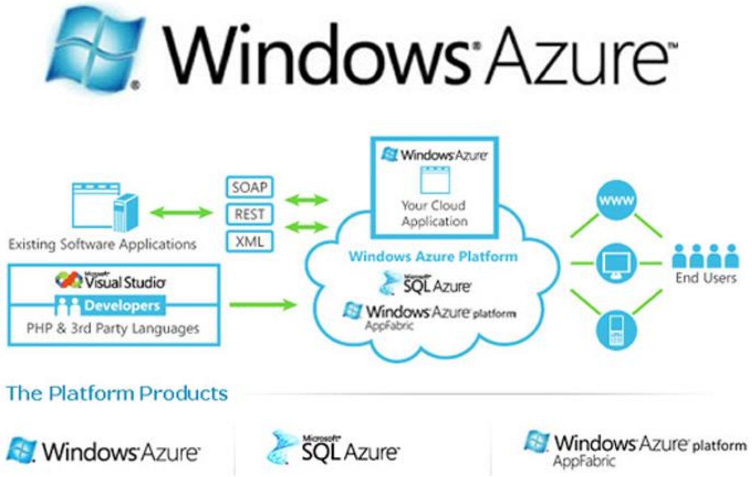
Development tools served up on a Platter a-la carte

No need to worry about upgrading to newer platforms or worry about license costs innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Windows Azure



What is Microsoft Azure?

- **Windows Azure**, which was later renamed as **Microsoft Azure in 2014**, is a **cloud computing platform**, designed by Microsoft to successfully build, deploy, and manage applications and services through a global network of data centers.

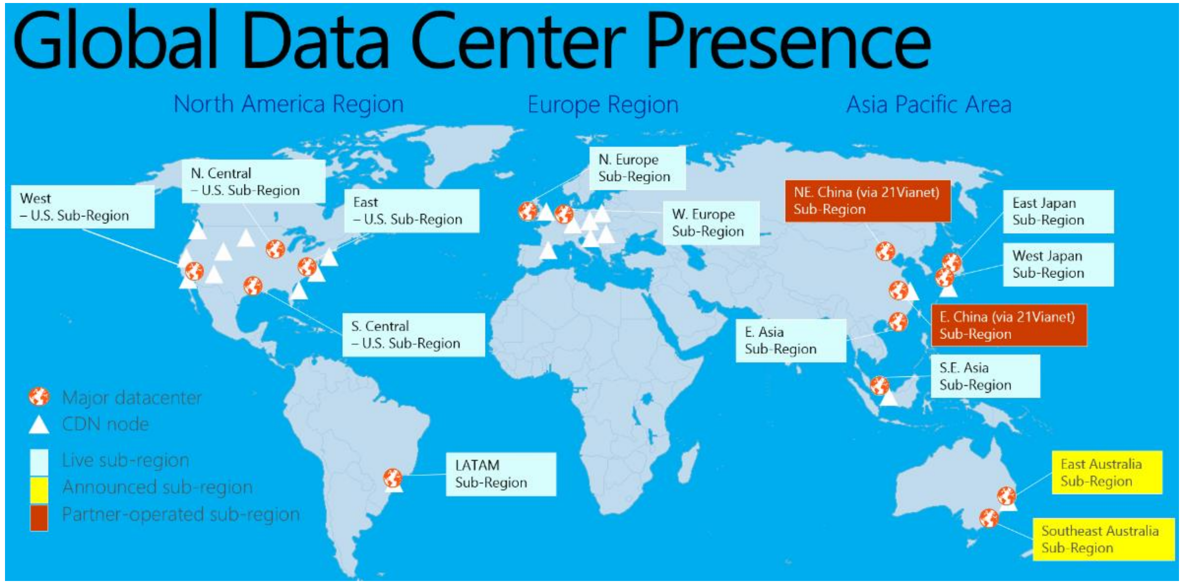
- Azure Platform is divided into three major component platforms. **Compute**, **Storage & Fabric Controller**.

- Azure can be described as the managed data centers that are used to build, deploy, manage the applications and provide services through a global network.

- The **services provided by Microsoft Azure** are **PaaS and IaaS**.

- Many programming languages and frameworks are supported by it. This platform is built over Microsoft data centers.

- Windows Azure can be used to create, distribute and upgrade Web applications without the need to maintain expensive, often underutilized resources onsite



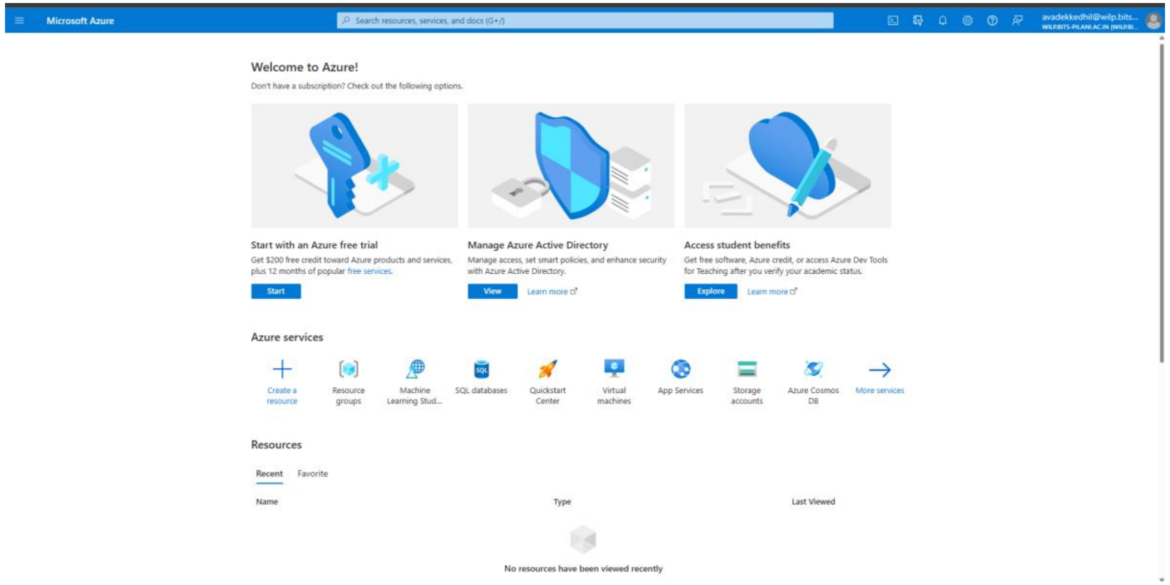
Windows Azure offers a cloud platform built on Windows OS and based on Microsoft virtualization technology.

Azure Components?

Azure provides more than **200 services**, are divided into **18 categories**.

These categories include **computing**, **networking**, **storage**, **IoT**, **migration**, **mobile**, **analytics**, **containers**, **artificial intelligence**, and other machine learning, integration, management tools, developer tools, security, databases, DevOps, media identity, and web services.

There are 42 Azure data centers spread around the globe, which is the highest number of data centers for any cloud platform. Also, Azure is planning to get 12 more data centers, which will increase the number of data centers to 54, shortly.



Microsoft Azure
Search resources, services, and docs (G+/)
avadekkedhil@wilp.bits...
WILPBITS-PILANI.AC.IN (WILPBI...

Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial

Get $200 free credit toward Azure products and services, plus 12 months of popular free services.

Manage Azure Active Directory

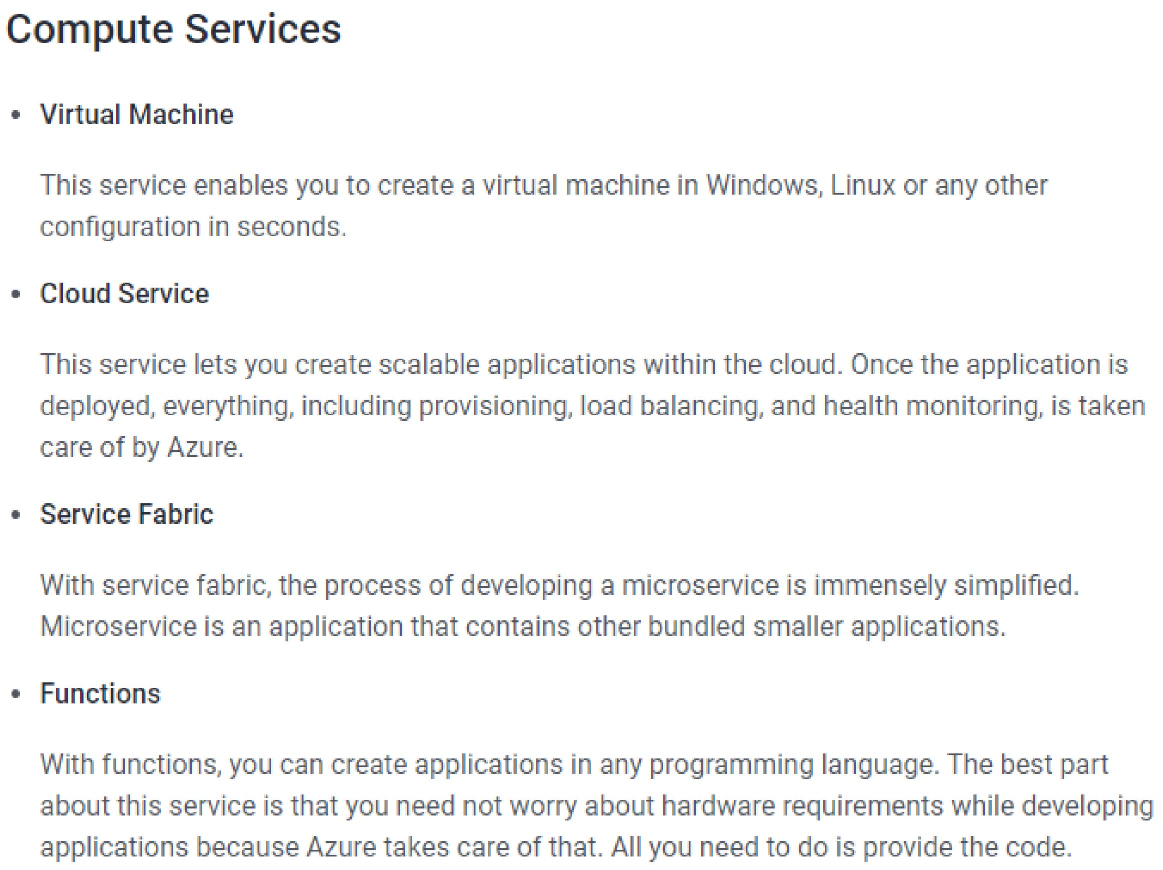
Manage access, set smart policies, and enhance security with Azure Active Directory.

Access student benefits

Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

No resources have been viewed recently

Azure Services?



Compute Services

- **Virtual Machine**

  This service enables you to create a virtual machine in Windows, Linux or any other configuration in seconds.

- **Cloud Service**

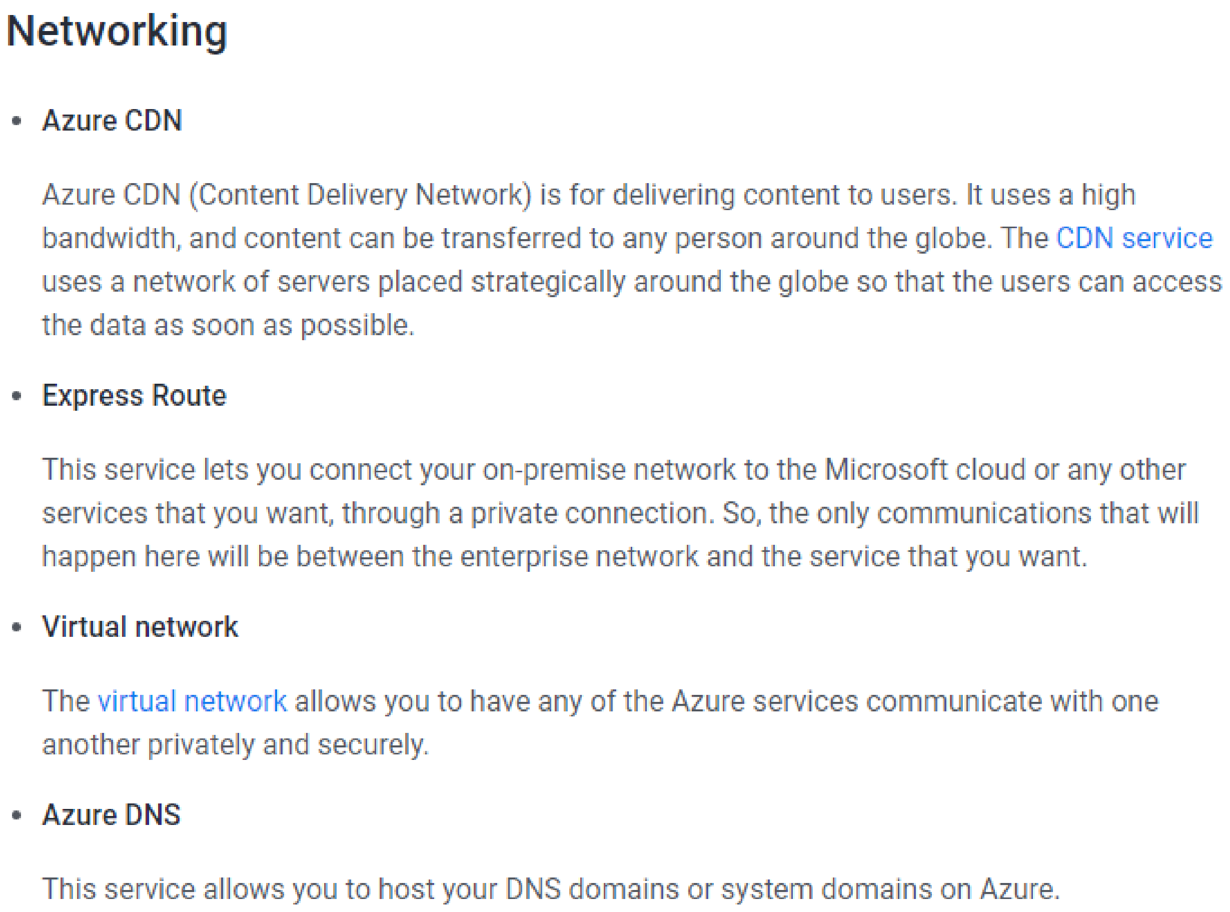
  This service lets you create scalable applications within the cloud. Once the application is deployed, everything, including provisioning, load balancing, and health monitoring, is taken care of by Azure.

- **Service Fabric**

  With service fabric, the process of developing a microservice is immensely simplified. Microservice is an application that contains other bundled smaller applications.

- **Functions**

  With functions, you can create applications in any programming language. The best part about this service is that you need not worry about hardware requirements while developing applications because Azure takes care of that. All you need to do is provide the code.



Networking

- **Azure CDN**

  Azure CDN (Content Delivery Network) is for delivering content to users. It uses a high bandwidth, and content can be transferred to any person around the globe. The CDN service uses a network of servers placed strategically around the globe so that the users can access the data as soon as possible.

- **Express Route**

  This service lets you connect your on-premise network to the Microsoft cloud or any other services that you want, through a private connection. So, the only communications that will happen here will be between the enterprise network and the service that you want.

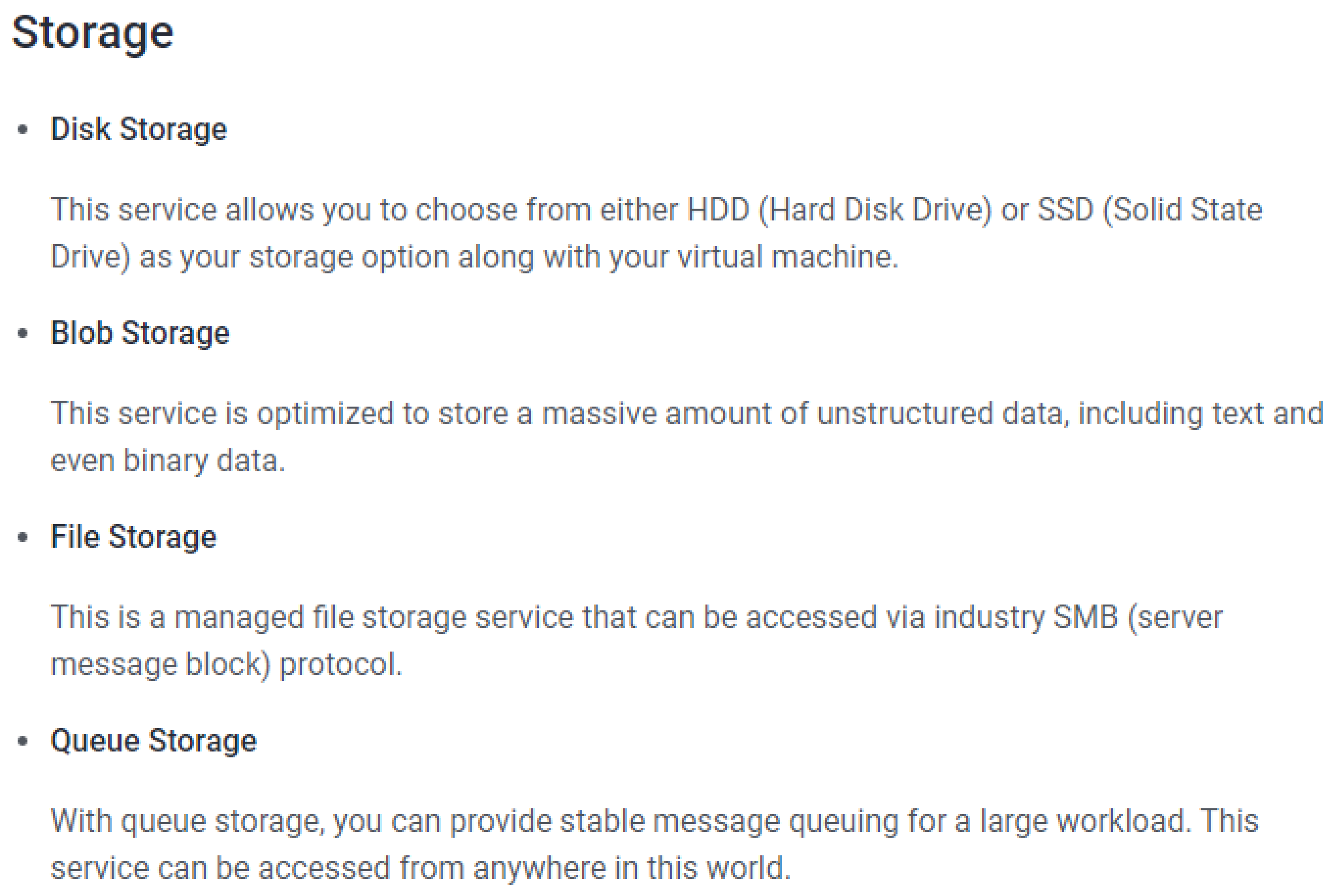
- **Virtual network**

  The virtual network allows you to have any of the Azure services communicate with one another privately and securely.

- **Azure DNS**

  This service allows you to host your DNS domains or system domains on Azure.

Azure Services?



Storage

- Disk Storage

  This service allows you to choose from either HDD (Hard Disk Drive) or SSD (Solid State Drive) as your storage option along with your virtual machine.

- Blob Storage

  This service is optimized to store a massive amount of unstructured data, including text and even binary data.

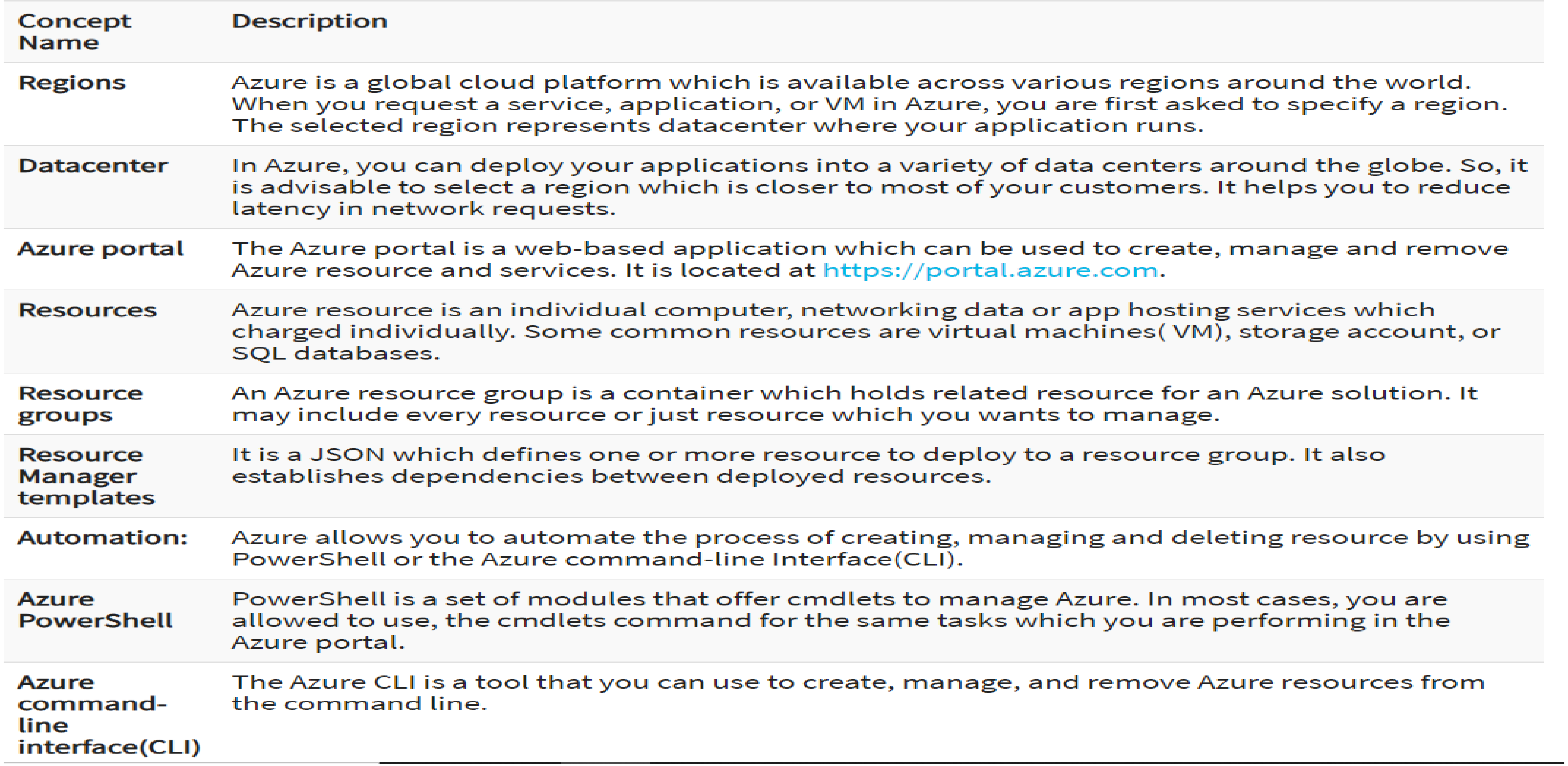
- File Storage

  This is a managed file storage service that can be accessed via industry SMB (server message block) protocol.

- Queue Storage

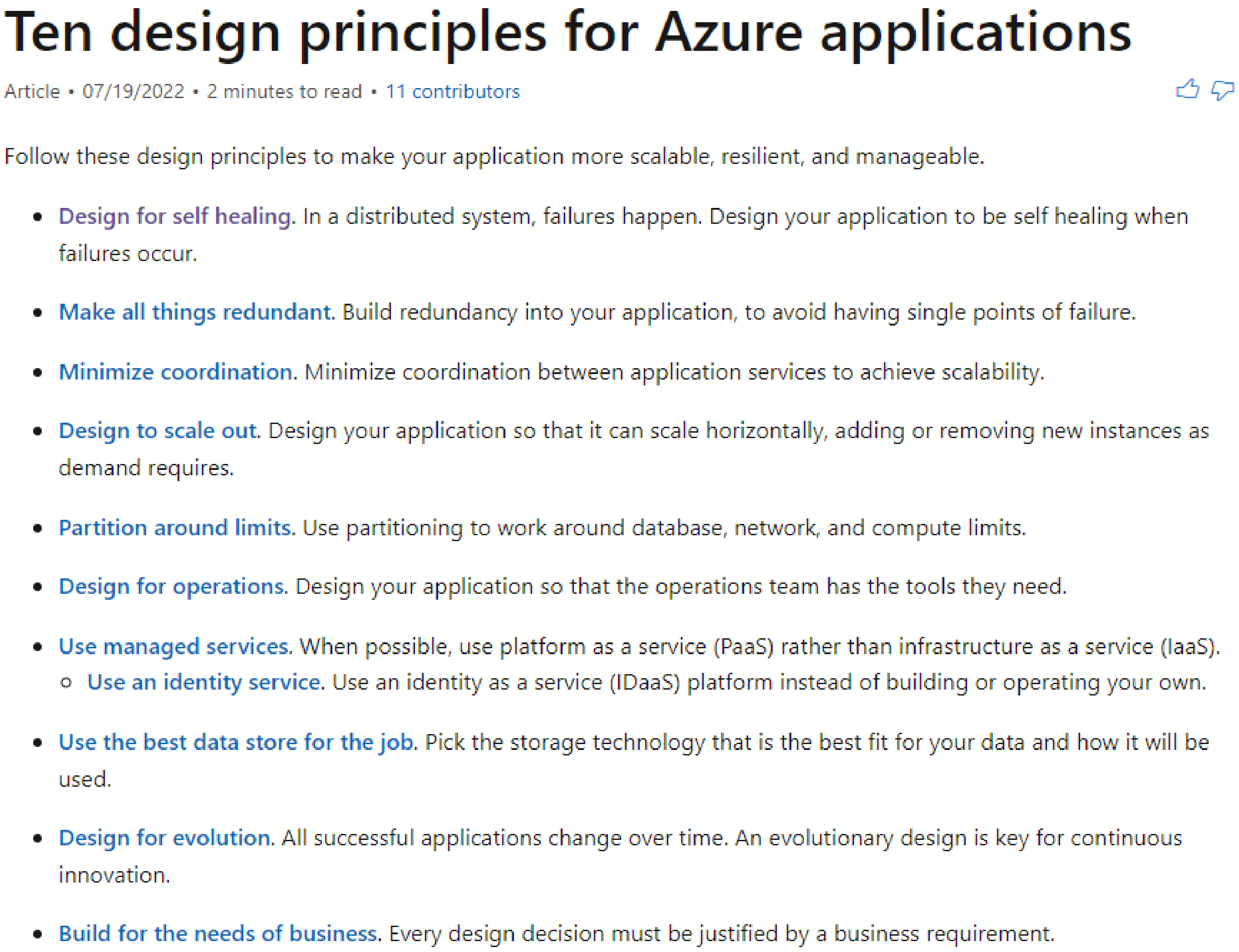
  With queue storage, you can provide stable message queuing for a large workload. This service can be accessed from anywhere in this world.

Azure key Terms?



Concept Name	Description
Regions	Azure is a global cloud platform which is available across various regions around the world. When you request a service, application, or VM in Azure, you are first asked to specify a region. The selected region represents datacenter where your application runs.
Datacenter	In Azure, you can deploy your applications into a variety of data centers around the globe. So, it is advisable to select a region which is closer to most of your customers. It helps you to reduce latency in network requests.
Azure portal	The Azure portal is a web-based application which can be used to create, manage and remove Azure resource and services. It is located at https://portal.azure.com.
Resources	Azure resource is an individual computer, networking data or app hosting services which charged individually. Some common resources are virtual machines(VM), storage account, or SQL databases.
Resource groups	An Azure resource group is a container which holds related resource for an Azure solution. It may include every resource or just resource which you wants to manage.
Resource Manager templates	It is a JSON which defines one or more resource to deploy to a resource group. It also establishes dependencies between deployed resources.
Automation:	Azure allows you to automate the process of creating, managing and deleting resource by using PowerShell or the Azure command-line Interface(CLI).
Azure PowerShell	PowerShell is a set of modules that offer cmdlets to manage Azure. In most cases, you are allowed to use, the cmdlets command for the same tasks which you are performing in the Azure portal.
Azure command-line interface(CLI)	The Azure CLI is a tool that you can use to create, manage, and remove Azure resources from the command line.

Azure PaaS Design Principles



Ten design principles for Azure applications

Article • 07/19/2022 • 2 minutes to read • 11 contributors

Follow these design principles to make your application more scalable, resilient, and manageable.

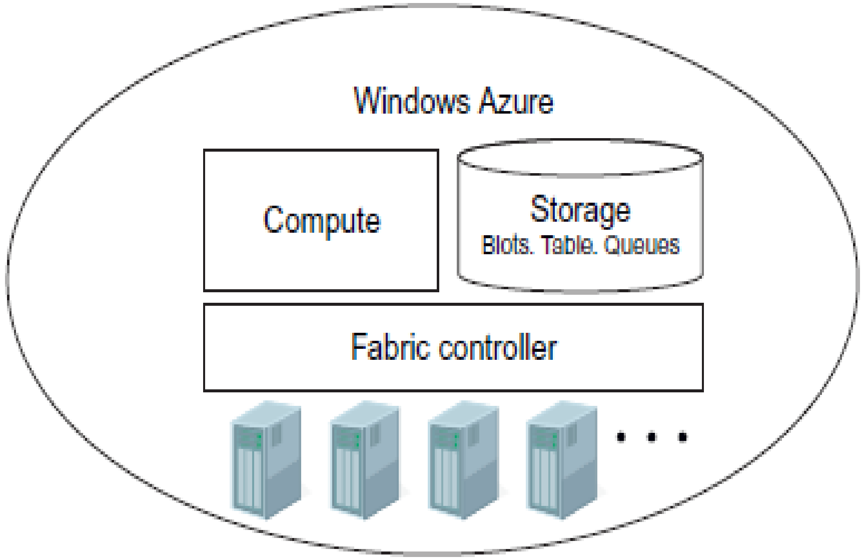
- Design for self healing. In a distributed system, failures happen. Design your application to be self healing when failures occur.
- Make all things redundant. Build redundancy into your application, to avoid having single points of failure.
- Minimize coordination. Minimize coordination between application services to achieve scalability.
- Design to scale out. Design your application so that it can scale horizontally, adding or removing new instances as demand requires.
- Partition around limits. Use partitioning to work around database, network, and compute limits.
- Design for operations. Design your application so that the operations team has the tools they need.
- Use managed services. When possible, use platform as a service (PaaS) rather than infrastructure as a service (IaaS).
  - Use an identity service. Use an identity as a service (IDaaS) platform instead of building or operating your own.
- Use the best data store for the job. Pick the storage technology that is the best fit for your data and how it will be used.
- Design for evolution. All successful applications change over time. An evolutionary design is key for continuous innovation.
- Build for the needs of business. Every design decision must be justified by a business requirement.

Azure Introduction

- Windows Azure provides a **platform to develop applications** using a range of available technologies and programming languages.
- It offers to create and deploy applications using .net platform, which is Microsoft's own application development technology. In addition to .net, there are many more technologies and languages supported. For example, Java, PHP, Ruby, Oracle, Linux, MySQL, Python.
- Windows Azure applications are scaled by creating **multiple instances** of the application.
- The **number of instances** needed by the **application is specified by the developer** while hosting the applications.
- If traffic is increased or decreased on the website or web application, it can be managed easily by logging in to Windows Azure management portal and specifying the instances. Load balancing can also be automated which would allow Azure to make the decision itself as when to assign more resources to application.
- Web applications support .net, java, python, php and node.js. Tasks such as scaling, and backups can be easily automated.
- A new feature called 'webjobs' is available, which is a kind of batch processing service. Webjobs can also be scaled and scheduled.
- The mobile application platforms supported are Xamarin iOS, Xamarin Android and IOS.
- Azure platform is developed in such a way that developers need to **concentrate on only the development part** and need not **worry about other technical stuff outside their domain**. Thus most of the administrative work is done by Azure itself.

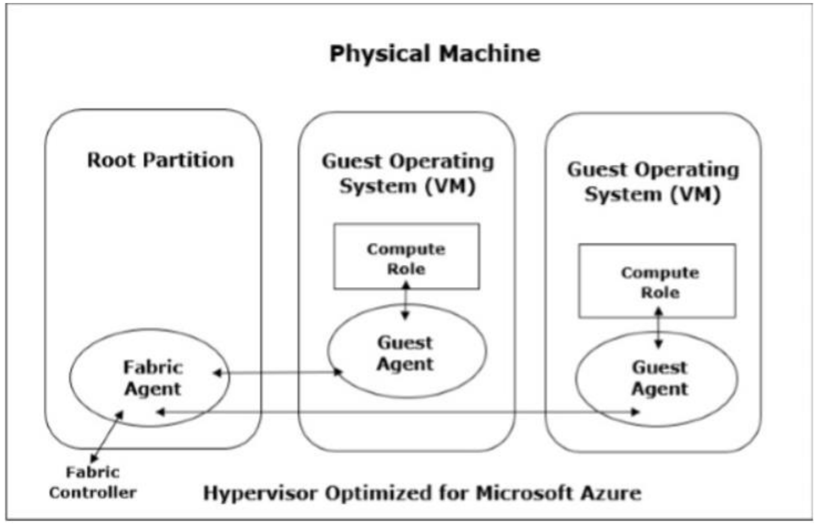
Azure Runtime Environment

- The Windows Azure runtime environment provides a scalable compute and storage hosting environment along with management capabilities. It has three major components: Compute, Storage and the Fabric Controller
- The hosting environment of Azure is called the **Fabric Controller**. It has a pool of individual systems connected on a network and automatically manages resources by load balancing and geo-replication. It manages the application lifecycle without requiring the hosted apps to explicitly deal with the scalability and availability requirements. Each physical machine hosts an Azure agent that manages the machine.
- The **Azure Compute Service** provides a Windows-based environment to run applications written in the various languages and technologies supported on the Windows platform.
- The Windows **Azure storage service** provides scalable storage for applications running on the Windows Azure in multiple forms. It enables storage for binary and text data, messages and structured data through support for features called Blobs, Tables, Queues and Drives.



Azure fabric Controller

- Fabric Controller is a significant part of Windows Azure architecture.
- Inside the data centre, there are many machines or servers aggregated by a switch. We can say that fabric controller is a brain of the azure service that analyses the processes and makes decisions.
- **Fabrics** are group of machines in Microsoft's data centre which are aggregated by a switch. The group of these machines is called **cluster**.
- Each cluster is managed and owned by a **fabric controller**. They are replicated along with these machines. It manages everything inside those machines, for e.g., load balancers, switches, etc. Each machine has a **fabric agent** running inside it and fabric controller can communicate with each fabric agent
- When a user chooses one of the virtual machine, the operating system, patch updates and software updates are performed by fabric controller. It decides where the new application should run which is one of the most important functions of Fabric Controller. It also selects the physical server to optimize hardware utilization
- When a new application is published in Azure, an application configuration file written in XML is also attached. The fabric controller reads those files in Microsoft datacenter and makes the setting accordingly



Imagine a situation where **four instances of web role** are running, and one of them dies.
The **fabric controller** will initiate a **new instance** to replace the dead one immediately.
Similarly, in case **any virtual machine fails**, a **new one is assigned by the fabric controller**. It also **resets the load balancers** after **assigning the new machine**, so that it points to the new machine instantaneously.
Thus, all the intelligent tasks are performed by the Fabric Controller in Windows Azure architecture.

Azure Components

- When the **system is running**, services are **monitored and one can access event logs, trace**/ debug data, performance counters, IIS web server logs, crash dumps, and other log files.

- **This information can be saved in Azure storage**. Note that there is **no debugging capability** for running cloud applications, but **debugging is done from a trace**.

- Like most PaaS services, Windows Azure defines a programming **model** specific to the platform, which is called the **Web role**- **Worker role** model.

- **Cloud Service Role**: In Azure, a Cloud Service Role is a collection of managed, load-balanced, Platform-as-a-Service virtual machines that work together to perform common tasks. Cloud Service Roles are managed by **Azure fabric controller** and provide the **ultimate combination of scalability, control, and customization**

- **Web Role** is a Cloud Service role in Azure that is configured and customized to **run web applications developed on programming languages** / technologies that are **supported by Internet Information Services** (IIS), such as ASP.NET, PHP, Windows Communication Foundation and Fast CGI

- **Worker Role** is any role in Azure that **runs applications and services level tasks**, which generally do not **require IIS**. In Worker Roles, IIS is not installed by default. They are mainly used to **perform supporting background processes** along with Web Roles and do tasks such as automatically compressing uploaded images, run scripts when something changes in database, get new messages from queue and process and more.

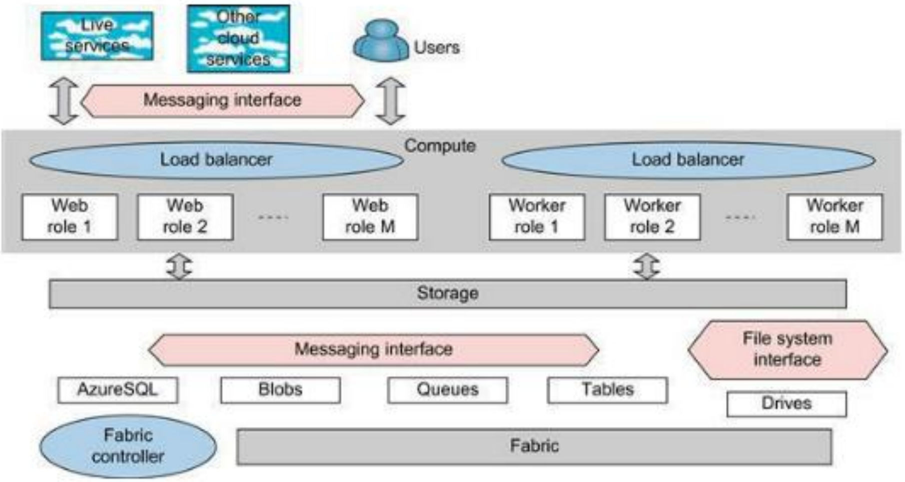
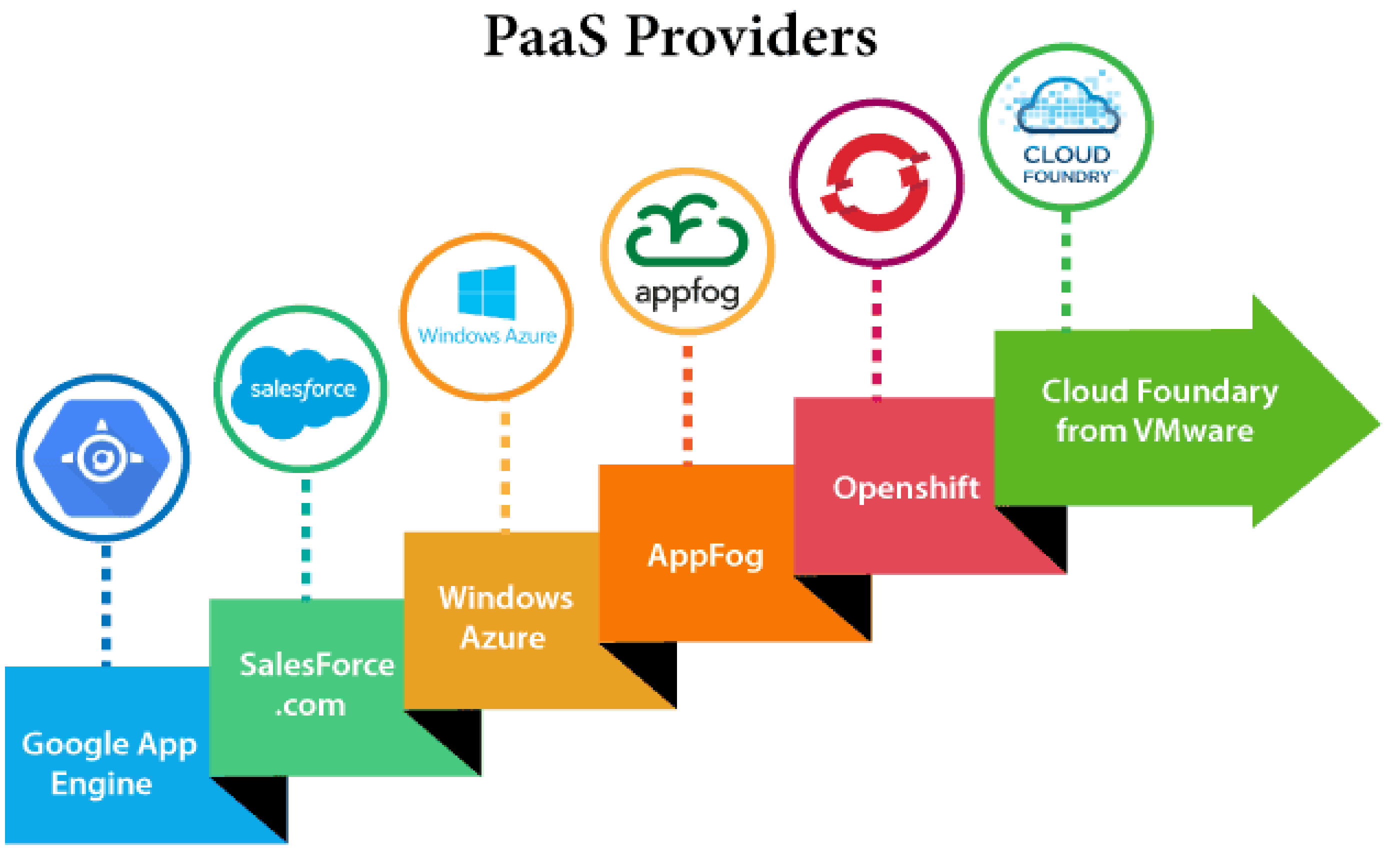


FIGURE 6.25 Features of the Azure cloud platform.

PaaS – Vendors (Popular)



Common PaaS vendors include Salesforce.com's Force.com, which provides an enterprise customer relationship management (CRM) platform.

PaaS platforms for software development and management include Appear IQ, Mendix, Amazon Web Services (AWS) Elastic Beanstalk, Google App Engine and Heroku.

Common PaaS vendors include Salesforce.com's Force.com, which provides an enterprise customer relationship management (CRM) platform.

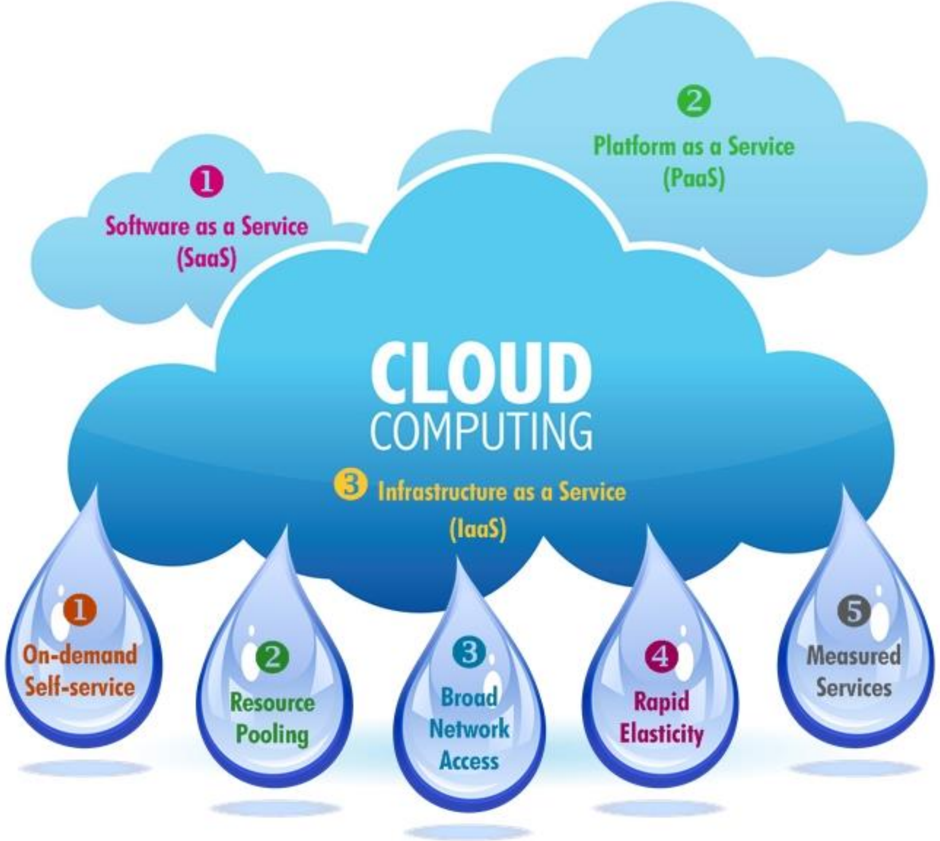
PaaS platforms for software development and management include Appear IQ, Mendix, Amazon Web Services (AWS) Elastic Beanstalk, Google App Engine and Heroku.

innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Software as a Service



Agenda

- What is SaaS?
- Traditional Model
- How is it delivered?
- SaaS Architecture
- SaaS Models
- Advantages of SaaS
- User and Vendor benefits of SaaS

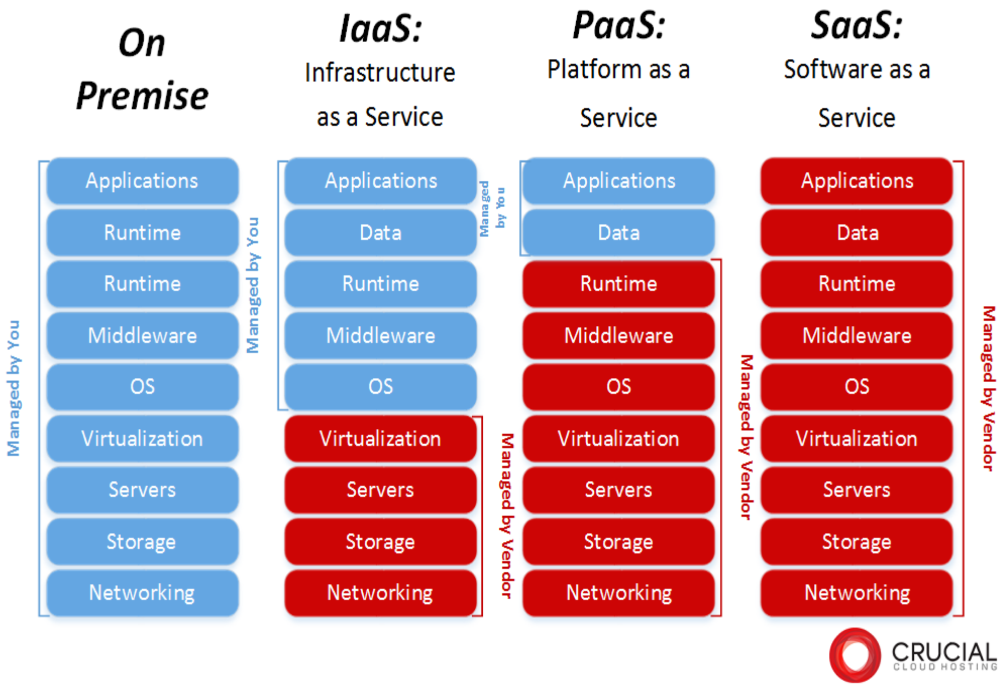
Introducing Software as a Service

- Software as a service is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet.

- Shortly, in the SaaS model software is deployed as a hosted service and accessed over the Internet, as opposed to “On Premise.”

- Software delivered to home consumers, small business, medium and large business

  The traditional model of software distribution, in which software is purchased for and installed on personal computers, is sometimes referred to as software as a product.



On Premise	*IaaS:* Infrastructure as a Service	*PaaS:* Platform as a Service	*SaaS:* Software as a Service
Applications	Applications	Applications	Applications
Runtime	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

SaaS Motivations

- In the traditional model of software delivery, the customer acquires a perpetual license and assumes responsibility for managing the software.

- There is a high upfront cost associated with the purchase of the license, as well as the burden of implementation and ongoing maintenance.

- ROI is often delayed considerably, and, due to the rapid pace of technological change, expensive software solutions can quickly become obsolete.

Traditional Software

Build Your Own

On-Demand Utility

Plug In, Subscribe

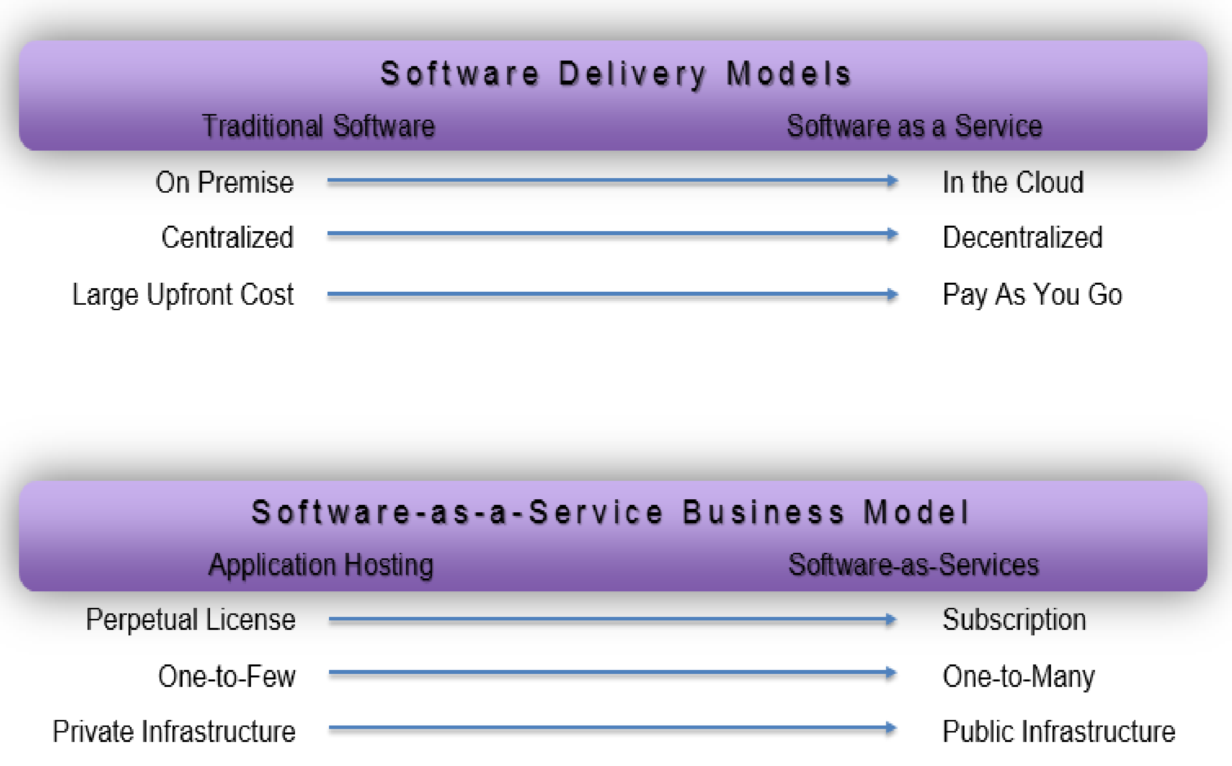
Pay-per-Use

SaaS Delivery Model

- The web as a platform is the centre point. The web as a platform is the centre point

- Network-based access to, and management of, commercially available (i.e., not custom) software application delivery that typically is closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics

- Software delivered to home consumers, small business, medium and large business
  - § The traditional model of software distribution, in which software is purchased for and installed on personal computers, is sometimes referred to as software as a product.



Software Delivery Models

Traditional Software	Software as a Service
On Premise	In the Cloud
Centralized	Decentralized
Large Upfront Cost	Pay As You Go

Software-as-a-Service Business Model

Application Hosting	Software-as-Services
Perpetual License	Subscription
One-to-Few	One-to-Many
Private Infrastructure	Public Infrastructure

SaaS Architecture

Run by

- Bandwidth technologies
- The cost of a PC has been reduced significantly with more powerful computing but the cost of application software has not followed
- Timely and expensive setup and maintenance costs
- Licensing issues for business are contributing significantly to the use of illegal software and piracy.

Scalable

- Multitenant efficient
- Configurable

Scaling the application - maximizing concurrency, and using application resources more efficiently

- i.e. optimizing locking duration, statelessness, sharing pooled resources such as threads and network connections, caching reference data, and partitioning large databases.

SaaS Architecture

Multi-tenancy – important architectural shift from designing isolated, single-tenant applications

- One application instance must be able to accommodate users from multiple other companies at the same time
- All transparent to any of the users.
- This requires an architecture that maximizes the sharing of resources across tenants
- is still able to differentiate data belonging to different customers.

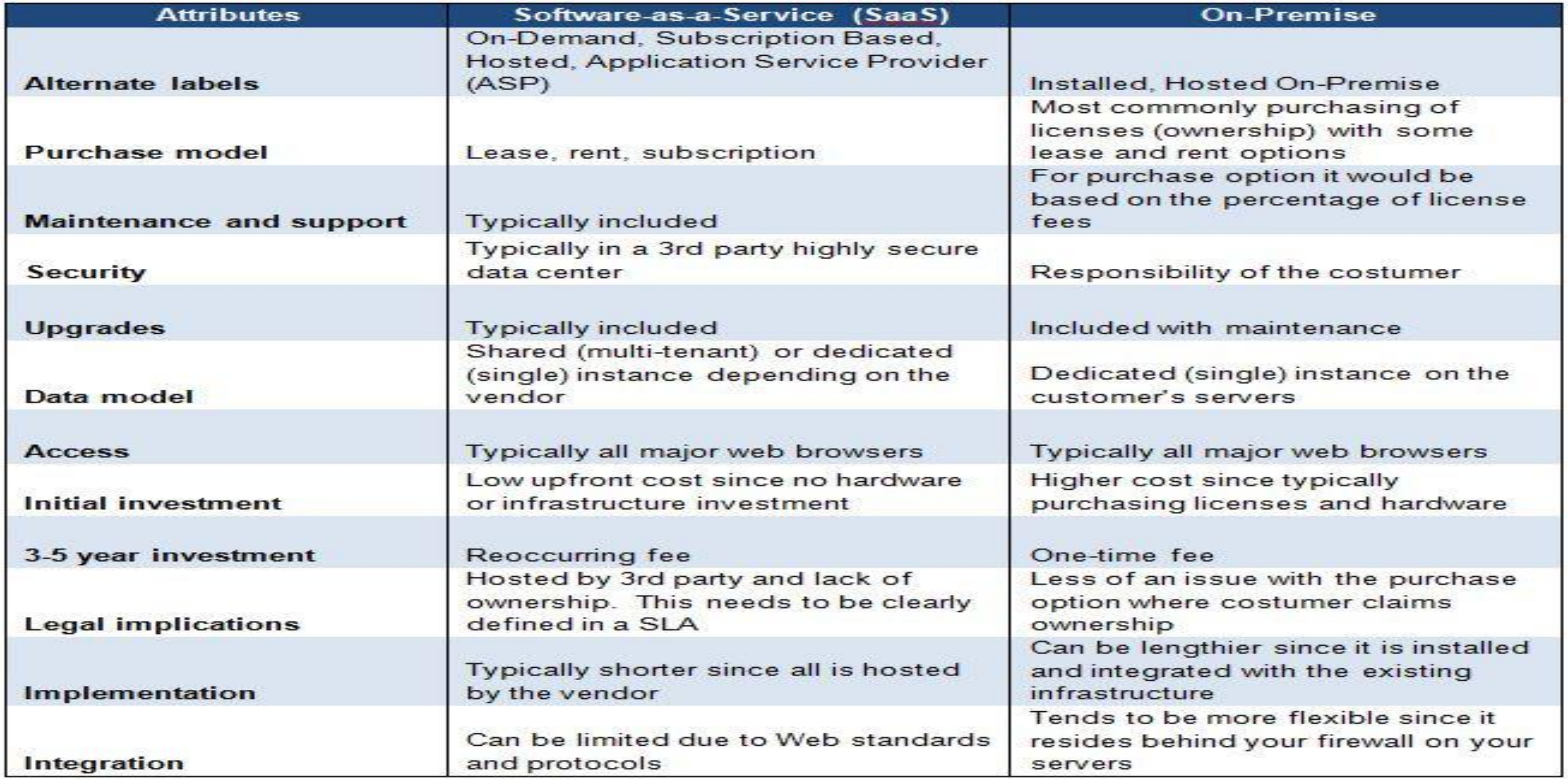
Configurable - a single application instance on a single server has to accommodate users from several different companies at once

- To customize the application for one customer will change the application for other customers as well.
- Traditionally customizing an application would mean code changes
- Each customer uses metadata to configure the way the application appears and behaves for its users.
- Customers configuring applications must be simple and easy without incurring extra development or operation costs

SaaS Model Comparison

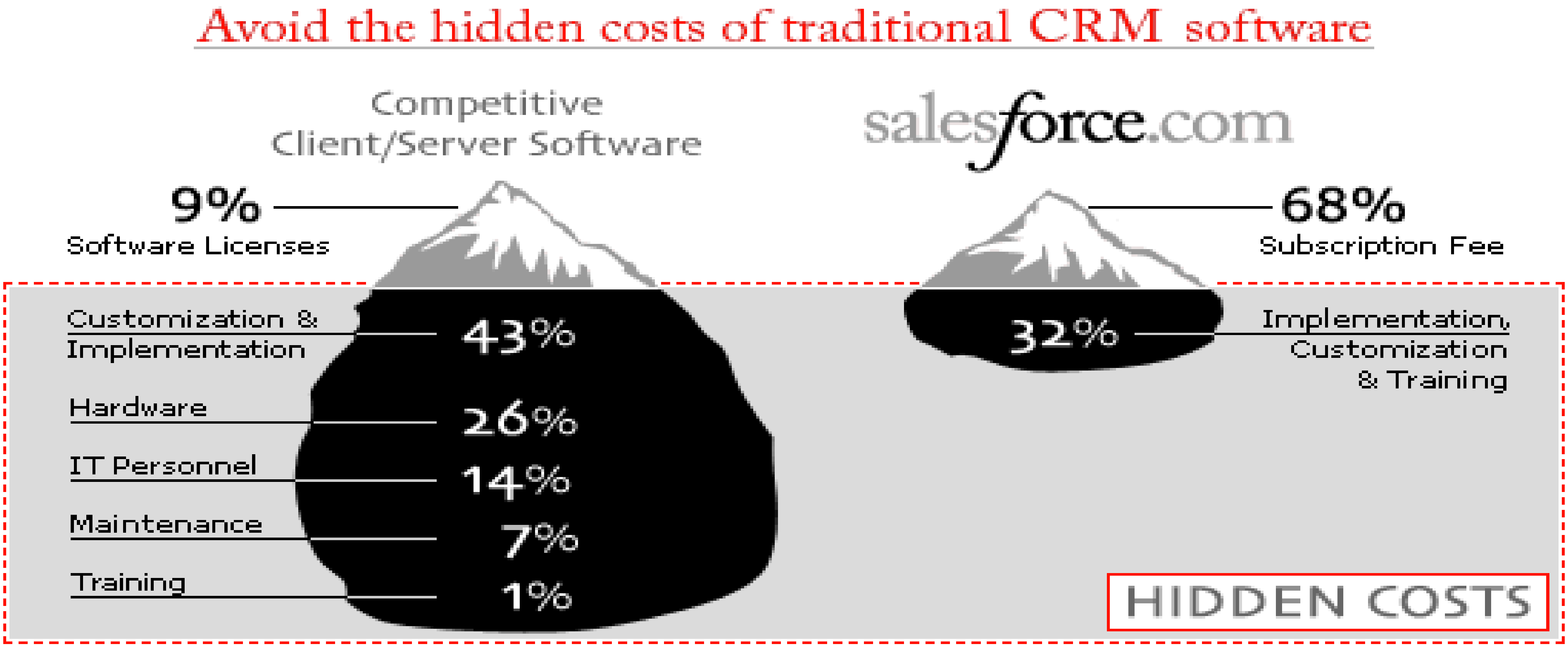
Traditional	Software as a Service
Designed for customers to install, manage and maintain.	Designed from the outset up for delivery as Internet-based services.
Architect solutions to be run by an individual company in a dedicated instantiation of the software.	Designed to run thousands of different customers on a single code.
Infrequent, major upgrades every 18-24 months, sold individually to each installed base customer.	Frequent, "digestible" upgrades every 3-6 months to minimize customer disruption and enhance satisfaction.
Version control Upgrade fee	Fixing a problem for one customer fixes it for everyone
Streamlined, repeatable functionality via Web services, open APIs and standard connectors	May use open APIs and Web services to facilitate integration, but each customer must typically pay for one-off integration work.

SaaS Model Comparison



Attributes	Software-as-a-Service (SaaS)	On-Premise
Alternate labels	On-Demand, Subscription Based, Hosted, Application Service Provider (ASP)	Installed, Hosted On-Premise
Purchase model	Lease, rent, subscription	Most commonly purchasing of licenses (ownership) with some lease and rent options
Maintenance and support	Typically included	For purchase option it would be based on the percentage of license fees
Security	Typically in a 3rd party highly secure data center	Responsibility of the costumer
Upgrades	Typically included	Included with maintenance
Data model	Shared (multi-tenant) or dedicated (single) instance depending on the vendor	Dedicated (single) instance on the customer's servers
Access	Typically all major web browsers	Typically all major web browsers
Initial investment	Low upfront cost since no hardware or infrastructure investment	Higher cost since typically purchasing licenses and hardware
3-5 year investment	Reoccurring fee	One-time fee
Legal implications	Hosted by 3rd party and lack of ownership. This needs to be clearly defined in a SLA	Less of an issue with the purchase option where costumer claims ownership
Implementation	Typically shorter since all is hosted by the vendor	Can be lengthier since it is installed and integrated with the existing infrastructure
Integration	Can be limited due to Web standards and protocols	Tends to be more flexible since it resides behind your firewall on your servers

SaaS Model Comparison



SaaS Adantages

Characteristics	Benefits
Network delivered access to commercially available software	**No local infrastructure or software to purchase or maintain** Applications & data are available anywhere with network connectivity
Application delivery is one-to-many model	Operating costs are reduced by managing infrastructure in central locations rather than at each customer's site
Built on optimized & robust platform	Improved availability and reliability
Customer pays for as much as they need when they need it	Lower TCO

VIRTUAL OFFICE

Google docs

Office 365

E-MAIL COMMUNICATION

Outlook.com

Gmail YAHOO! MAIL

STORAGE

SaaS Providers

1. Lumen5
2. FutureFuel
3. Squibler
4. Buffer
5. Dropbox
6. AWS
7. HubSpot
8. Cloud-Based Microsoft Office 365
9. Salesforce
10. G Suite
11. ZenDesk
12. Visme
13. Slack
14. Canva
15. Box

SaaS Providers at a Glance

SaaS Providers
Lumen5 is a leading video creator SaaS app that lets businesses create amazing videos with its drag-and-drop interface. It creates video automatically form text or any URL. The text is converted into a video that can be personalized by positioning text, adding images from the library, highlighting keywords, adding brand colors, tweaking font style, and changing video resolution. Lumen5 receives 220K+ monthly visitors which shows how popular it is
FutureFuel is a perfect SaaS example that is sure to inspire you. It is a student debt management app for businesses and HR practitioners. It helps companies attract and retain top talent by addressing repaying the loans of students. The students, in return, work for their company as dedicated employees. FutureFuel provides businesses with access to the future workforce that will stick with them for years to come.
Squibler is a SaaS creativity app that helps writers tell stories. It has a great interface that makes writing hassle-free. It has many prompt writing categories that serve all types of writing such as books, novels, journals, screenwriting, or others.
Buffer is a perfect SaaS example for pretty much any SaaS business out there. It is a simple yet effective tool that helps social media management and scheduling easier for businesses of all sizes. But what’s more important is that how Leo Widrich guest blogged for 10 months which helped Buffer acquire 100K users in less than a year.

SaaS Providers at a Glance

SaaS Providers
Dropbox is a leading cloud storage SaaS company that makes it easier for businesses to store, share, and collaborate on files and data on the go. It offers you a smart workplace that lets your workforce work from anywhere.
Amazon Web Services (AWS) as a SaaS example. With more than 150 services on offer, AWS provides businesses and individuals with all the tools they need such as database, IoT, business applications, machine learning, storage, robotics, security, customer engagement, blockchain, and more.
HubSpot is a leading SaaS tool for businesses and even enterprise software companies. It offers sales, marketing, CRM, CMS, and services software to businesses. The marketing software helps businesses attract visitors, convert them, close deals, and retain them.
Cloud-based Office 365 is indeed something worth talking about. You can now create, edit, share, manage, and access your office files from any device. Office 365 cloud is a pure SaaS example that shows how Microsoft has fulfilled the needs of its users by offering them SaaS products. You can still install and use Microsoft office 365 on your personal computer but with the cloud platform, you get access to it on the go as you can access it from any Microsoft's data center.

SaaS Providers at a Glance

SaaS Providers
Salesforce It is one of the most popular software as a service provider that excels in cloud computing. It is a complete customer relationship management suite for businesses. Salesforce lets businesses collect, store, access, monitor, and analyze customer data from a single dashboard.
The **G Suite** is the Google Cloud SaaS applications that include several cloud-based apps. G Suite includes Gmail, Calendar, Hangouts, Google Drive, Sheets, Docs, Forms, Slides, Sites, Vault, and several other apps.
ZenDesk is another example of a SaaS company that is indeed inspiring. It is a SaaS company with annual revenue of more than $5 million. It is a customer support and ticketing software that supports several support channels including phone, email, like chat, social media, online tickets, and more.
Box is a cloud content management and file-sharing that is publicly traded SaaS company. It is a multi-purpose software for businesses that lets them collaborate, automate, share, and manage content and files over the cloud. It supports secure file sharing and files can be accessed from desktop, mobile, and web. Team members can collaborate and discuss everything in real-time on the document they are working on.

SaaS User Benefits

- **Lower Cost of Ownership**
  - The software is paid when it is consumed, no large upfront cost for a software license Salesforce.com has a best-of-breed CRM system for $59.00 per user per month, with no upfront
  - Since no hardware infrastructure, installation, maintenance, and administration, budgeting is easy
  - The software is available immediately upon purchasing
- **Focus on Core Competency**
  - The IT saving on capital and effort allows the customer to remain focused on their core competency and utilize resources in more strategic areas.
- **Access Anywhere**
  - Users can use their applications and access their data anywhere they have an Internet connection and a computing device
  - This enhances the customer experience of the software and makes it easier for users to get work done fast
- **Freedom to Choose (or Better Software)**
  - The pay-as-you-go (PAYG) nature of SaaS enables users to select applications they wish to use and to stop using those that no longer meet their needs. Ultimately, this freedom leads to better software applications because vendors must be receptive to customer needs and wants.

SaaS User Benefits

- **New Application Types**
  - Since the barrier to use the software for the first time is low, it is now feasible to develop applications that may have an occasional use model. This would be impossible in the perpetual license model. If a high upfront cost were required the number of participants would be much smaller.
- **Faster Product Cycles**
  - Product releases are much more frequent, but contain fewer new features than the typical releases in the perpetual license model because the developer know the environment the software needs to run
  - This new process gets bug fixes out faster and allows users to digest new features in smaller bites, which ultimately makes the users more productive than they were under the previous model.

SaaS Vendor Benefits

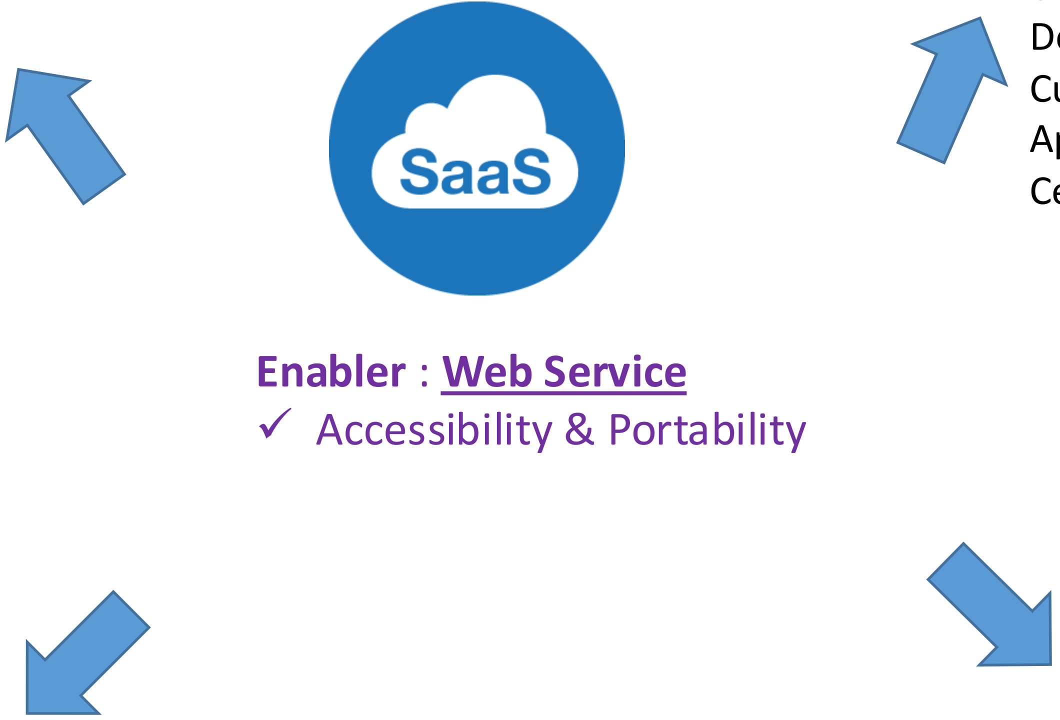
- **Increased Total Available Market**
  - Lower upfront costs and reduced infrastructure capital translate into a much larger available market for the software vendor, because users that previously could not afford the software license or lacked the skill to support the necessary infrastructure are potential customers.
  - A related benefit is that the decision maker for the purchase of a SaaS application will be at a department level rather than the enterprise level that is typical for the perpetual license model. This results in shorter sales cycles.
- **Enhanced Competitive Differentiation**
  - The ability to deliver applications via the SaaS model enhances a software company's competitive differentiation. It also creates opportunities for new companies to compete effectively with larger vendors.
  - On the other hand, software companies will face ever-increasing pressure from their competitors to move to the SaaS model.
  - Those who lag behind will find it difficult to catch up as the software industry continues to rapidly evolve.
- **Lower Development Costs & Quicker Time-to-Market**
  - The main saving is at testing (35%). Small and frequent releases – less to test
    - Application is developed to be deployed on a specific hardware infrastructure, far less number of possible environment – less to tes
    - This, in turn, provides the software developer with overall lower development costs and quicker time-to-market.

SaaS Vendor Benefits

- **Effective Low Cost Marketing**
  - Between 1995 and today, buyers' habits shifted from an outbound world driven by field sales and print advertising to an inbound world driven by Internet search.
- **Predictable MRR Revenue**
  - Traditionally, software companies rely on one major release every 12-18 months to fuel a revenue stream from the sale of upgrades (long tail theory).
  - In the SaaS model the revenue is typically in the form of Monthly Recurring Revenue (MRR)
- **Improved Customer Relationships**
  - SaaS contributes to improved relationships between vendors and customers.
- **Protecting of IP**
  - Difficult to obtain illegal copies
  - Price is low, making getting an illegal copies totally unnecessary

Software as a Service

Why SaaS



Capability

Saas provides the following
- Hosted , Finished Product
- Subscription to Services

Characteristics

Not owned, but subscribed to from an external service provider.
Designed to be Multi Tenant
Customer Configuration instead of Application configuration
Centralized management

Enabler : **Web Service**

✓ Accessibility & Portability

Models

SaaS can be obtained as
(1) Managed Service
(2) Monthly Subscription

Benefit

No need to purchase Software or Licenses upfront
Centralized management helps SaaS Vendors
Lower TCO for users

SaaS Applicability Scenarios

1 Single-User software application

- Organize personal information
- Run on users' own local computer
- Serve only one user at a time
- **Inapplicable to SaaS model**
  - Data security issue
  - Network performance issue
- Example: Microsoft office suite (Prior to o365)

2 Infrastructure software

- Serve as the foundation for most other enterprise software application
- **Inapplicable to SaaS model**
- Installation locally is required
- Form the basis to run other application
- Example: Window XP, Oracle database

3 Embedded Software

Software component for embedded system

Support the functionality of the hardware device

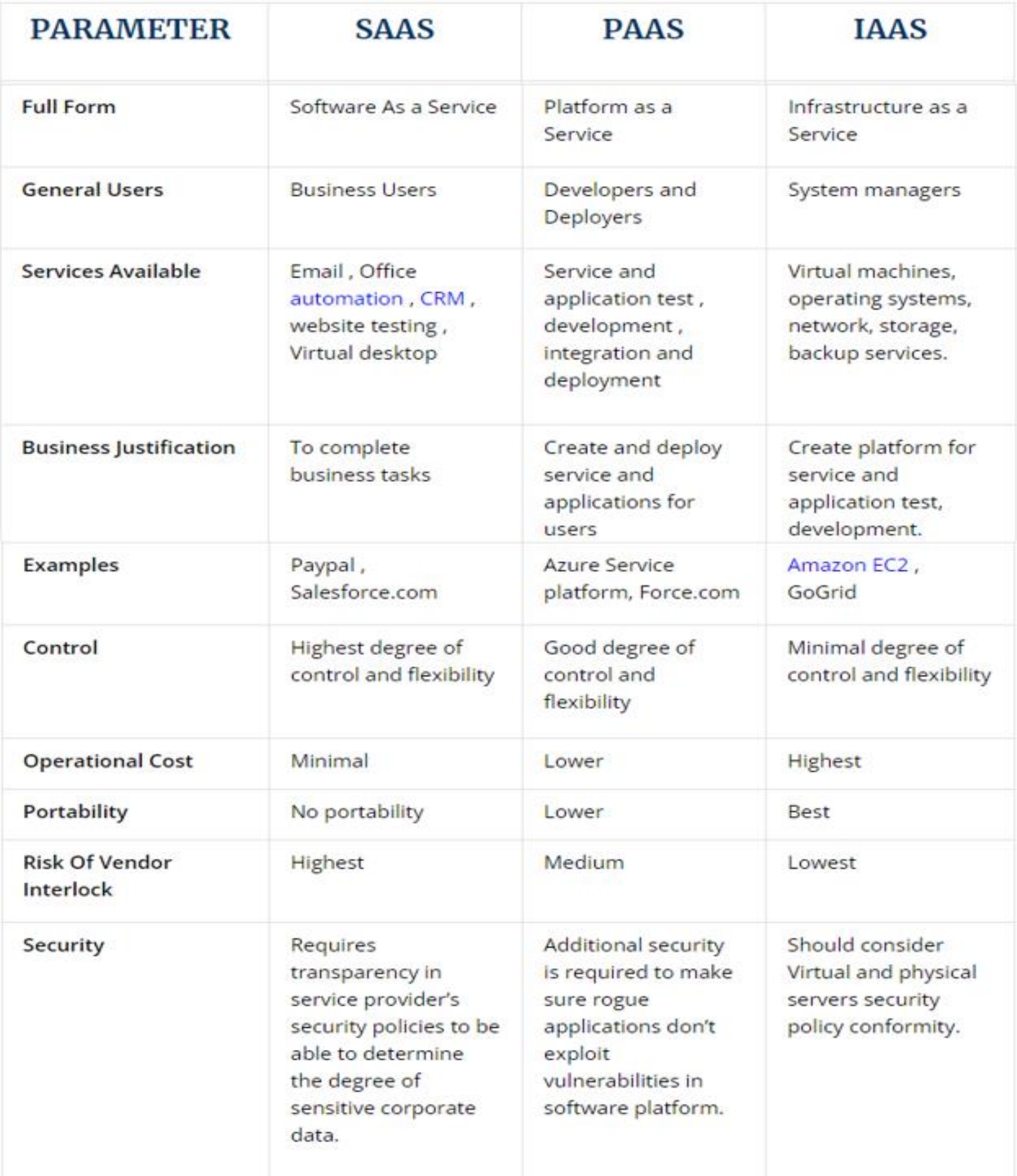
Inapplicable to SaaS model

Embedded means software and hardware is combined together and is inseparable like a bios / firmware

Example: software embedded in ATM machines, cell phones, routers, medical equipment, etc

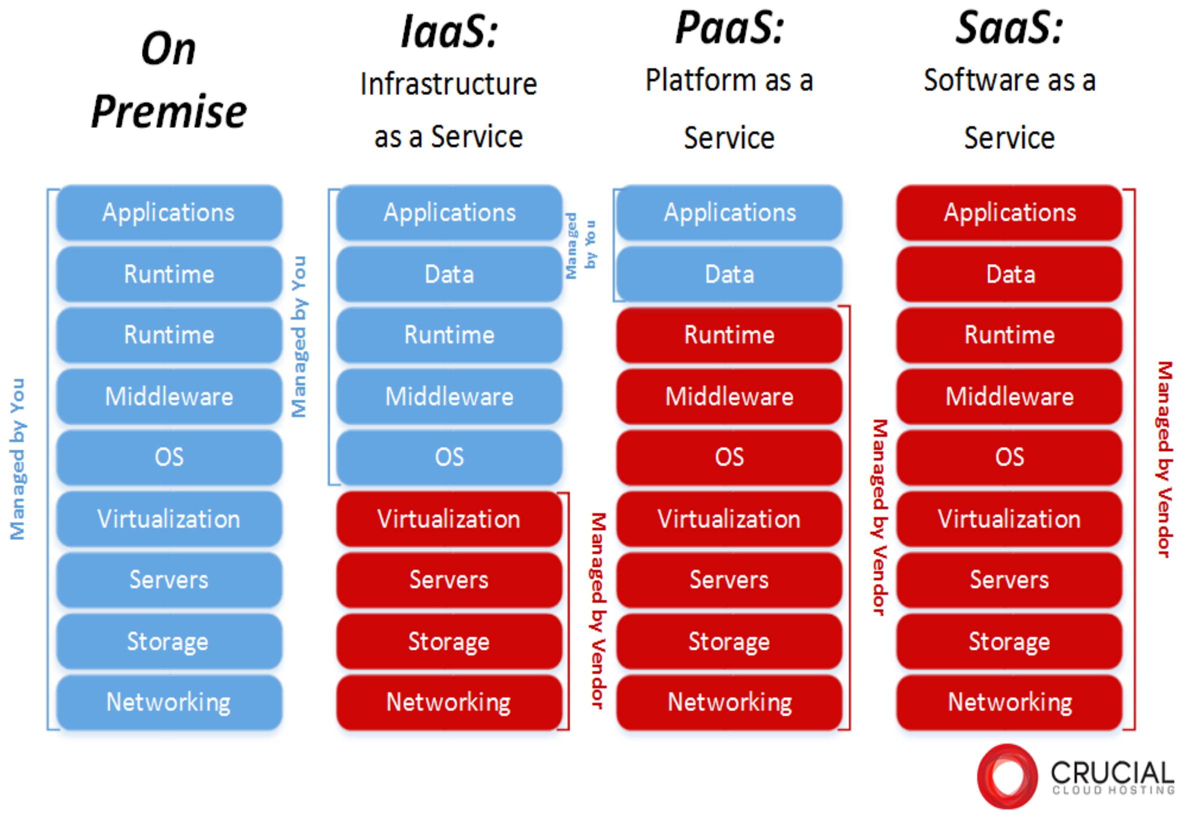
The Right SaaS Model?

- **Enterprise Software Application**
  - Perform business functions
  - Organize internal and external information
  - Share data among internal and external users
  - The most standard type of software applicable to SaaS model
  - Example: Saleforce.com CRM application, Siebel On-demand application

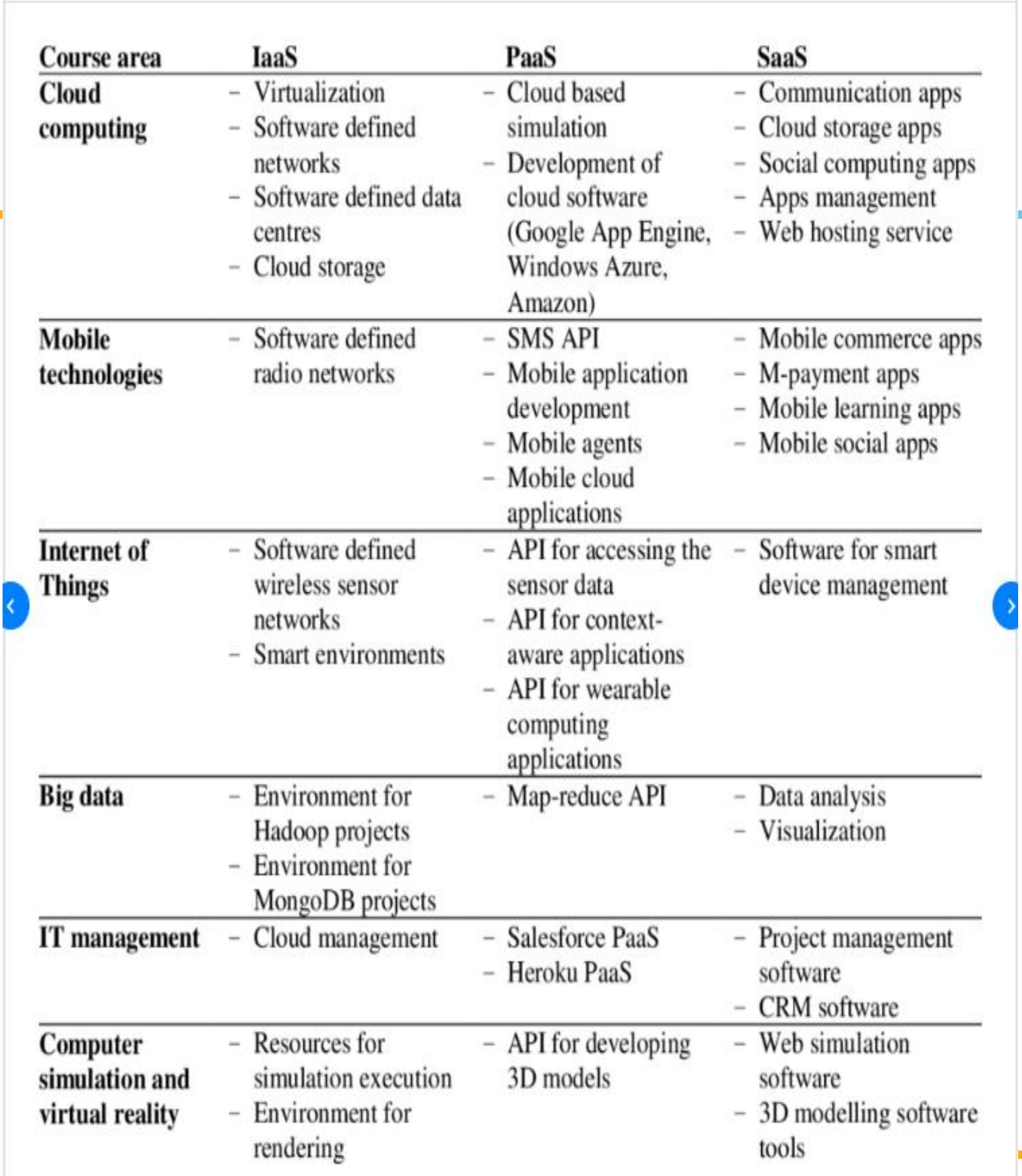


PARAMETER	SAAS	PAAS	IAAS
Full Form	Software As a Service	Platform as a Service	Infrastructure as a Service
General Users	Business Users	Developers and Deployers	System managers
Services Available	Email , Office automation , CRM , website testing , Virtual desktop	Service and application test , development , integration and deployment	Virtual machines, operating systems, network, storage, backup services.
Business Justification	To complete business tasks	Create and deploy service and applications for users	Create platform for service and application test, development.
Examples	Paypal , Salesforce.com	Azure Service platform, Force.com	Amazon EC2 , GoGrid
Control	Highest degree of control and flexibility	Good degree of control and flexibility	Minimal degree of control and flexibility
Operational Cost	Minimal	Lower	Highest
Portability	No portability	Lower	Best
Risk Of Vendor Interlock	Highest	Medium	Lowest
Security	Requires transparency in service provider's security policies to be able to determine the degree of sensitive corporate data.	Additional security is required to make sure rogue applications don't exploit vulnerabilities in software platform.	Should consider Virtual and physical servers security policy conformity.

SaaS vs PaaS vs IaaS

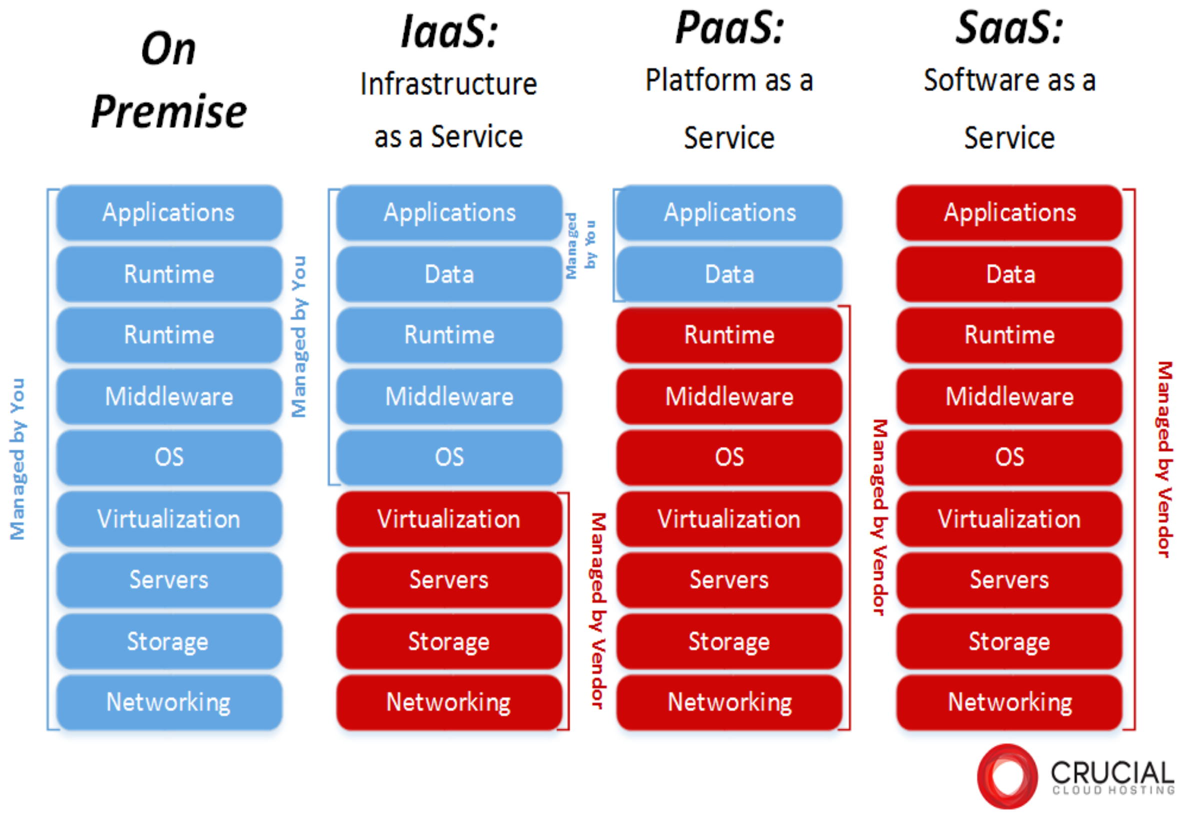


COMPARISION



Course area	IaaS	PaaS	SaaS
Cloud computing	– Virtualization – Software defined networks – Software defined data centres – Cloud storage	– Cloud based simulation – Development of cloud software (Google App Engine, Windows Azure, Amazon)	– Communication apps – Cloud storage apps – Social computing apps – Apps management – Web hosting service
Mobile technologies	– Software defined radio networks	– SMS API – Mobile application development – Mobile agents – Mobile cloud applications	– Mobile commerce apps – M-payment apps – Mobile learning apps – Mobile social apps
Internet of Things	– Software defined wireless sensor networks – Smart environments	– API for accessing the sensor data – API for context-aware applications – API for wearable computing applications	– Software for smart device management
Big data	– Environment for Hadoop projects – Environment for MongoDB projects	– Map-reduce API	– Data analysis – Visualization
IT management	– Cloud management	– Salesforce PaaS – Heroku PaaS	– Project management software – CRM software
Computer simulation and virtual reality	– Resources for simulation execution – Environment for rendering	– API for developing 3D models	– Web simulation software – 3D modelling software tools

SaaS vs PaaS vs IaaS



On Premise

Applications, Runtime, Runtime, Middleware, OS, Virtualization, Servers, Storage, Networking — Managed by You

IaaS: Infrastructure as a Service

Applications, Data, Runtime, Middleware, OS — Managed by You

Virtualization, Servers, Storage, Networking — Managed by Vendor

PaaS: Platform as a Service

Applications, Data — Managed by You

Runtime, Middleware, OS, Virtualization, Servers, Storage, Networking — Managed by Vendor

SaaS: Software as a Service

Applications, Data, Runtime, Middleware, OS, Virtualization, Servers, Storage, Networking — Managed by Vendor

CRUCIAL CLOUD HOSTING

APPLICABILITY

Q & A……..

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Credits

- Hwang, Kai; Dongarra, Jack; Fox, Geoffrey C.. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things (Kindle Locations 3532-3533). Elsevier Science. Kindle Edition.